Understanding and exploiting user intent in community question answering

Understanding and Exploiting User Intent in Community Question Answering

Long Chen

Submitted in partial fulfillment of the

requirements for the degree

of Doctor of Philosophy

in the Department of Computer Science & Information Systems

BIRKBECK, UNIVERSITY OF LONDON

April 2014

Declaration

This thesis is the result of my own work, except where explicitly acknowledged in the text.

Long Chen _____

Abstract

A number of Community Question Answering (CQA) services have emerged and proliferated in the last decade. Typical examples include Yahoo! Answers, WikiAnswers, and also domain-specific forums like StackOverflow. These services help users obtain information from a community — a user can post his or her questions which may then be answered by other users. Such a paradigm of information seeking is particularly appealing when the question cannot be answered directly by Web search engines due to the unavailability of relevant online content. However, question submitted to a CQA service are often colloquial and ambiguous. An accurate understanding of the intent behind a question is important for satisfying the user's information need more effectively and efficiently.

In this thesis, we analyse the intent of each question in CQA by classifying it into five dimensions, namely: subjectivity, locality, navigationality, procedurality, and causality. By making use of advanced machine learning techniques, such as Co-Training and PU-Learning, we are able to attain consistent and significant classification improvements over the state-of-the-art in this area. In addition to the textual features, a variety of metadata features (such as the category where the question was posted to) are used to model a user's intent, which in turn help the CQA service to perform better in finding similar questions, identifying relevant answers, and recommending the most relevant answerers.

We validate the usefulness of user intent in two different CQA tasks. Our first application is question retrieval, where we present a hybrid approach which blends several language modelling techniques, namely, the classic (query-likelihood) language model, the state-of-the-art translation-based language model, and our proposed intent-based language model. Our second application is answer validation,

where we present a two-stage model which first ranks similar questions by using our proposed hybrid approach, and then validates whether the answer of the top candidate can be served as an answer to a new question by leveraging sentiment analysis, query quality assessment, and search lists validation.

Contents

List of Figures

List of Tables

Acknowledgments

I would like to take this favourable opportunity to express my sincere gratitude to all the people who have made a contribution to my research. It wouldn't have been possible to write this doctoral thesis without my friends around me, who helped me in their myriad ways. Unfortunately, only a small proportion of them are able to be mentioned in the following due to space limitation.

Above all, I would like to thank my parents, who have provided me their committed support and infinite patience throughout my PhD. My brother has given me his faithful support, as always, for which the verbal expression of thanks does not suffice. This thesis would not have been possible without the guidance, support, and patience of my principal supervisor, Dr. Dell Zhang, let alone his constructive advices and profound knowledge on Information Retrieval and Machine Learning. He showed me how an impressive piece of work can be done in a swift and orderly fashion; he often got to the issue with a single pertinent remark, which I failed to identify at first but immediately recognised to be the crux of the problem. I would also like to thank the advice, trust and support from my second supervisor, Professor. Mark Levene. He gave me his trust by allowing me to carry out research all by myself (which is crucial for my career development), yet he provided me his support and advice whenever I needed them.

Thanks to all those, at Birkbecks Computer Science department and at the London Knowledge Lab. They supplied a great place and facility for me to work. Thanks in particular to Tony Lewis. More than any other, in the first two years, he brightened up my daily life with some small talks on language and culture, I shall reminisce our old days of playing Jujitsu and juggling together, he is my first teacher for both of them.

Thanks to all those at Clandon House, who provided me with a family friendly environment. Thanks to, Sneha Krishnan, and Aniset Kamanga, more

than any others, they enriched my daily life by going shopping and dining once in a while. Thanks in particular to Tom Ue, who helped me out with his meticulous proofreading for several pieces of my work. Perhaps more importantly, his well-intentioned harassment on the daily basis makes my PhD life considerably less painful. Thanks to Sofi Qi and Adam Summerfield, who hung out with me every Saturday afternoon on various topics, which has deepened my understanding of the difference between Chinese and British culture. I shall miss those beautiful British afternoons.

Thanks to the community of researchers in the Community Question Answering. I'm so lucky to be able to work in an excellent community of researchers, who work on one of the most important issues of research at present. Thanks to the immense body of programmers who have collaboratively contributed to todays open source projects. The projects which are employed, directly and indirectly, in the thesis is countless but the software list which worth a mentioning here includes: Java, C++, R, Matlab, Python, Weka, and LaTeX.

To all these people: Thanks a lot!

Chapter 1

Introduction

The World Wide Web (Web) provides a very large-scale and dynamic hyperspace of information. In recent years, some Web 2.0 style CQA services have been released, which allow a user to post his or her questions which may then be answered by other users. Through interaction with the online community, knowledge can easily be transferred between users of different background. Typical examples of CQA services includes Yahoo! Answers, Wiki Answers, Quora, Baidu Zhidao and also domain-specific forums such as StackOverflow. Such a paradigm of information seeking is particularly appealing when the user's information need cannot be satisfied directly by Web search engines or automatic QA systems (there is no real QA systems available on the web apart from Wolfram Alpha). Furthermore, compared with computer algorithms used in automated QA (see Section 1.2), humans have a better ability of understanding natural language so other users are often able to give more relevant and comprehensive results to complex information needs expressed as natural language questions than Web search engines are. Last but not least, CQA services often directly contribute to search engines by publishing their content – questions and associated answers – to the Web, and making them indexable by search engines, so as to allow search engine users (given that a new

query is submitted) to find answers directly by reusing previously asked questions. However, despite the progress that has been made, there is still a large margin for improvement in many perspectives of CQA services, some of which include:

1. The quality of the questions and answers, in general, are not satisfactory.
2. Question recommendation mechanisms have not yet been implemented in many CQA services, such as Yahoo! Answers, and these questions cannot always be resolved by the most pertinent candidates.
3. Current question searches do not have good support regarding the question of complex information needs (for example, questions usually have certain temporal or geographical restrictions).
4. Keeping regular users active is a challenging task. (For example, as we mentioned that regular users in Quora are not active enough due to its quality control mechanism.)
5. Keeping the expert users active is also very difficult. The biggest challenge with the current CQA design is that the expert users earn many points too easily to the extent that most of them do not care about earning points anymore.
6. Current question searches usually fail to consider users' emotion and subjectivity. (As the case of the previous example: "Why do Americans ask questions assuming that they are the only people on earth?").

To begin with, this chapter will step through several typical Web applications which include: Web Search Engines, Automatic Question Answering. Then we formally introduce the problems we are tackling and the thesis contribution, which respectively describe the focus of this work and the contribution of this work in

the immense body of literature. We close this chapter with thesis outline, which summarize how we are going to address differing problems in each chapter.

1.1 Web Search Engines

Search engine technology is at the heart of the Web, for they have redefined the way for people to seek and interact with information. They have become ubiquitous in our daily life with the proliferation of the mobile Web, which is accessible to a large number of mobile phone users.

Most search engines comprise three major components: the crawler, the index and the search-engine software. A crawler (or spider) is a program that visits URLs (Uniform Resource Locators) from link to link and copies their Web pages and other information. Everything the spider copies from the web goes into an index, which is the second component of a search engine. Indices keep files stored on servers connected to the Internet, which help search engines to find the relevant web pages in a much shorter time and at a dramatically lower cost. The last component, namely search engine software, is responsible for searching Web pages in the indices (which contain the query terms submitted by a user), and ranking the relevant web pages by their weights (which are calculated by a variety of factors, such as the term frequency in the document). With the aforementioned three components working in synergy, modern search engines are capable of finding any conceivable information about people, events, news, and a myriad of other information in fractions of a second.

However, there are several challenges that current search engines need to overcome:

1. An unprecedentedly large repository of information is accumulated, which is composed of over 30 trillion documents from a variety of sources. It is a

difficult task for search engine to model users' information need on the basis of such a massive scale with users of varying interests and backgrounds.

2. The information seeking paradigm of search engines usually fail to satisfy complex information need in the format of colloquial or verbose queries. In light of this, queries submitted to search engines are usually very short — for example, the average query length of the Excite search engine log in 2001 is 2.4 words[1]. It is, therefore, an extremely hard task for users to accurately formulate their complex information needs into just a few keywords.

3. Search engines may not be able to find relevant web pages for some queries whose information have not been publicised at a website as yet.

4. Given that search engines have returned the desired information successfully, users still need to read through the results list to pinpoint the relevant content, which may involve tedious work for users to find what is truly needed (For instance, in Google, up to the first 1000 results can be shown with 10 displayed per page).

1.2 Automatic Question Answering

To address the above challenges, automatic Question Answering (QA) systems have been developed with the aim to directly deliver clear and concise answers to a new question in a timely manner. Next generation search engines integrate automatic QA systems by understanding the question and summarising knowledge from the large-scale datasets. Some preliminary automatic QA systems have participated in QA track in the Text Retrieval Conference (TREC), the most influential QA competition organized by the National Institute of Technology (NIST) (see, for ex-

[1] http://en.wikipedia.org/wiki/Web_search_query

ample, Voorhees [76, 77]). The first type of questions that researchers have looked into were factoid questions. For example, "where was X born?", "When did Y take place?" Later on, researchers also aimed to handle more complex types of questions. Some typical examples are biographical questions such as "Who is Albert Einstein?"; definitional questions such as "What is Higgs Boson?"; and list questions such as "List the universities located in London." Each year TREC releases a test set which consists of several hundred questions and an evaluation system which assesses the answers submitted to the automatic systems. TREC then ranks each systems results in terms of either MRR (Mean Reciprocal Rank, which is the multiplicative inverse rank of the first correct answer) or accuracy (the percentage of correctly answered questions). Despite the impressive results reported by some researchers, the majority of the community can only produce a mediocre performance. That is, there are no standard models which are capable of producing an accuracy higher than 50% on the TREC test sets [77]. Research on automatic QA is still an active research area on the going. For example, Etzioni et al. [23] endeavor to advance automatic QA by improving information extraction techniques. They introduced the "next generation search engine" (Open Information Extraction)[2] based on open-domain information extractors. The system makes a single data-driven pass over the corpus containing billions of web pages, and extracts millions of relational assertions without requiring human labelling process. Despite its advanced nature compared to current search engines, the Open Information Extraction system strictly limits the question format in the syntactic pattern as "who/what verb who/what", which largely reduces the contribution margin towards natural language support. In light of this, it is evident that there are still many unresolved issues in the research of automatic QA.

START[3], the world's first online automatic QA system, was developed in

[2] http://openie.cs.washington.edu/
[3] http://start.csail.mit.edu/

1993 and has been operating until now. However, the system is only capable of delivering answers to questions about places (e.g., cities, countries, and coordinates), movies (e.g., titles, actors, and directors), and people (e.g., birth dates and biographies). The most influential online automatic QA system is arguably Ask Jeeves (known as Ask.com), which was formally founded in 1996. Ask Jeeves has unveiled a dataset consisting of 300 million questions, aiming to provide users with more accurate result.

Unfortunately, Ask Jeeves can only achieve limited success in the QA field. For example, it simply pulls results from various search engines if it fails to answer a question (which is nothing new compared to the other major search engines). In that sense, Ask Jeeves is not a fully automated QA system as yet.

More recently, Wolfram Research has also developed its own online automatic QA system namely, Wolfram Alpha[4], which is a successful commercial answer engine. Wolfram Alpha provides real-time services that can resolve factual questions directly by extracting the answer from external resources — instead of retrieving a list of web pages as the case of the typical search engines. The external resources are derived from both academic and commercial websites, which includes the CIA's World Factbook[5] and the United States Geological Survey[6]. Even though Wolfram Alpha works remarkably well for answering questions of computational facts (for example, "who is the first American president?" or more complex questions such as "How old was President Reagan when he died?"), it is not capable of answering questions topics related to social sciences and cultural studies. It also does not support factual questions which require a narrative response such as "What's the difference between an alligator and a crocodile?"

In 2011 the debut of IBM Watson, an artificial intelligence system devel-

[4] http://www.wolframalpha.com/
[5] https://www.cia.gov/library/publications/the-world-factbook/
[6] http://www.usgs.gov/

oped in IBM's DeepQA project, has attracted much attention[7]. Watson indexed a large amount of web page content, which consumes four terabyte of disk storage, including the full text of Wikipedia. Watson competed on Jeopardy! (a TV quiz show) against human players, from which it received the first prize of one million dollars in 2011. In the game, Watson consistently outperformed its human rivals but had difficulties in responding to a few topics with short clues of only a few words. Despite its success in the game and some other domain specific areas (such as management decisions for medical utilization), Watson still only has a limited power in answering nonrestrictive, real-time questions — for example, it is still unable to distinguish what is socially appropriate language.

In summary, despite the progress that has been made, there are still many unresolved problems in automatic QA:

1. Understanding natural language is an extremely difficult task, which requires immense progress in natural language processing and knowledge representation and inference.

2. Even for factual question answering, short phrases or sentences are often not informative enough to resolve the question.

3. A Majority of real world questions comprise complex information needs (for example, questions usually have certain temporal or geographical restrictions), which go beyond the capacity of the current automatic QA systems.

4. The answers of some questions may not be available on the Web, as we have already mentioned (in Section 1.1), which can only be resolved by the power of humans.

5. For many questions there is no standard answer, as in the case of opinionated

[7] http://www.research.ibm.com/labs/watson/index.shtml

questions ("Why do Americans ask questions assuming that they are the only people on earth?").

1.3 Problem Definition

Understanding the intent behind a new question is a natural direction for improving CQA services, since it can supply users with more personalized, and more effective CQA services tailored to their information needs. For example, we may want to employ different strategies to answer questions with different intent. However, current research on user intent in search engines cannot be directly applied to CQA services.

In CQA users normally ask natural language questions, which are addressed to humans, whereas in Web search users submit keyword queries which are addressed to computerised algorithms. More specifically, this leads to the following five major differences between CQA questions and search engine queries:

1. Many CQA questions are inherently subjective. It has been shown that the proportion of Yahoo! Answers oriented to factual question answering is decreasing while subjective/complex question answering is gradually increasing [50].

2. Many CQA questions are socially motivated, as users know that the answers to their questions would be coming from other users in the community. Instead of satisfying an information need, such questions are actually about establishing social connections (e.g., finding a date), or about generating some empathy (e.g., complaining), or just for entertainment purposes (e.g. telling jokes).

3. Even though about 10% of queries submitted to search engines are in question format [22], they are quite different from the question patterns used in CQA

services. For example, instead of using the common question format "What is a", or "Where is" in CQA, question queries in search engines are more likely to be the formats as "I need", "I want", "Show me".

4. CQA questions are more likely to have additional constraints, since they are usually longer and more complex than the search engine queries. For example, people may ask something in a specific area (e.g., looking for restaurants), or within a specific time frame (e.g., seeking for news).

5. Compared with search engines, CQA services have richer information, which can be used to characterise one's social status. For instance, each user has their unique asking and answering history; each question may correspond to a best answer, and an upvote/downvote value; furthermore, some user may have the pattern of asking questions in several specific topics (e.g., Traveling). This kind of rich information can help CQA system to reveal the user intent by providing evidence from the user's perspective, in addition to the surface textual features from the questions themselves.

Furthermore, even though there have been CQA studies, which investigate strategies for one or two dimensions of the user intents, they mostly summarise each question as a clear and simple information need (so that the computer can understand it easily). Question answering systems are required to understand the user intent at a deeper level. In this thesis we investigate potential answers to the following three questions regarding user intent in CQA:

- *How to categorise different user intents in CQA? (taxonomy)*
- *How to automatically identify the user intents of a question from a CQA service? (classifier)*

- *How to incorporate the user intents to improve the performance of CQA services? (e.g., question retrieval and answer validation)*

Investigating all these questions form a picture depicting the multi-dimensional nature of the user intent would help us not only to understand the question more deeply but also in a broader context.

1.4 Thesis Contribution

The three-fold contribution of this thesis can be summarised as follows:

1. We identify user intents from a user-centric perspective, for which we classify questions into five (user intent) dimensions with the aim of the deep understanding of the search goal. A simple definition regarding those dimensions can be found in Table 1.1.

2. We develop advanced classification techniques, which are capable of utilising both a variety of metadata features (such as the category where the question was posted to) and the surface textual features, to model users' intents.

3. We exploit user intents (which we learned from the classification) to find similar questions and identify similar answers, which in turn help to improve the performance of CQA services.

1.5 Thesis Outline

Chapter 2 reviews background research on Community Question Answering, from the basics of a CQA service, to classical approaches for question retrieval, question classification, answer recommendation and answer validations. The chapter closes

Table 1.1: The simple definition for each question dimension.

intent	definition
OSS	The intent of such questions is to get knowledge, opinions, or social interaction
locality	The intent of such questions is to get information of a certain locality.
navigational	Navigational questions are those whose answer can be resolved by web pages.
procedural	How-to-questions are those whose answer is a set of procedures.
causal	Why-questions are those whose answer is a causative description.

with a statistical summary of the datasets used, which are the foundation for several experiments conducted in this thesis.

Chapter 3 begins by describing *objective/subjective/social intent* from a user-centric perspective, for which we classify questions into three categories according to their underlying user intent: subjective, objective, and social. Our investigation reveals that textual features and metadata features are conditionally independent of each other, and each of them is sufficient for prediction. Therefore they can be exploited as two views in Co-Training (a semi-supervised learning framework) to make use of a large amount of unlabelled questions, in addition to the small set of manually labelled questions, for enhanced question classification. The user intent (objective/subjective/social) of each candidate question is predicted by a probabilistic classifier which makes use of both textual features and metadata features.

Chapter 4 introduces the *locality intent*, in which questions are classified into two categories according to their intent scope: local and global. The challenge for this task is that manually labelling questions as local or global for training would be very costly. Realising that we could find many local questions reliably from a few location-related categories (e.g., "Travel"), we propose to build local/global question classifiers in the framework of PU-Learning (i.e., learning from positive and

unlabelled examples), and thus remove the need of manually labelling questions. In addition to standard text features of questions, we also make use of locality features which are extracted by a geo-parsing tool, such as Yahoo! Placemaker. Our experiments on real-world datasets (collected from Yahoo! Answers and WikiAnswers) show that the probability estimation approach at PU-Learning outperforms other proposed approaches, S-EM (spy EM) and Biased-SVM for this task.

Chapter 5 analyses the *navigational intent*, in which questions are classified as navigational and non-navigational. We define questions that are resolved (or largely explained) by the linked web pages (i.e., in the corresponding answers) as navigational questions, which are simulated as verbose queries to evaluate the performance of search engines (i.e., by considering the associated linked web pages as relevant documents). We then experiment with the process of identifying new navigational questions from CQA, from which we demonstrate that navigational intent detection can be effectively automated by using textual features and a set of metadata features.

Chapter 6 describes *procedural intent*, in which we identify a series of empirical patterns to identify how-to-questions and estimate the probability whether a new how-to question in CQA, such as Yahoo! Answers, can be satisfactorily answered by the external resource using a two-stage model similar to factual question answering. A broad range of techniques spanning from query quality assessment to search list validation are leveraged to extract features for our model. A classifier with the features modelling the question context (e.g., the categories where the question was posted) is compared to the surface text and query feedback of the question.

Chapter 7 tackles the problem of using *causal intent* to help users to receive product reviews. In addition to the technique of query quality assessment and search lists validation, it also incorporates some other techniques for feature

generation, such as sentiment analysis and lexico-syntactics.

Chapter 8 demonstrates the utility of the above mentioned user intents through a hybrid approach to question retrieval that blends several language modelling techniques for question retrieval, namely, the classic (query-likelihood) language model, the translation-based language model (an approach similar to query expansion, which is capable of addressing the lexical gap problem), and our proposed intent-based language model.

Chapter 9 finishes this thesis by providing a summary of the contributions and the conclusions of each chapter. Several future directions are then discussed, regarding alternative approaches for improving several components of the framework, as well as directions for extending the framework for other information seeking behaviours.

Chapter 2

Related Work

Although the history of CQA is quite short, it has already attracted a large amount of interest from researchers, spanning from information seeking behavior [81], resources comparison [28], question recommendation [73] to user intent [27]. Current research on CQA services entails studying the user's background, motives, and methods by which people seek and share their information. It may also involve system development for supporting such activities.

Considering the thesis aims to understand users' intents by harnessing machine learning techniques, it is important to understand the position of this thesis within the immense body of literature on CQA. This chapter will step through several typical CQA systems in Section 2.1. Then we will discuss different approaches to the understanding and exploiting of user intent, namely question classification, question retrieval, answer validation, and answer recommendation. Section 2.2 covers the literature relevant to the use of question classification. Section 2.3 covers the literature relevant to question retrieval and ways of measuring relevant questions. Section 2.4 covers the literature relevant to the use of answer validation. Section 2.5 covers the literature relevant to question recommendation. Section 2.6 describes the literature relevant to user intent understanding in the context of Web search.

Section 2.7 closes with the statistics of CQA datasets used in this thesis.

2.1 Community Question Answering

Considering the limited success of the current automatic QA systems, another attractive way for resolving a question is by making use of the wisdom of the crowds, also known as "collective intelligence." Such social systems are called Community Question Answering (CQA).

CQA services usually consists of three components [70]: first, a mechanism which allows users to submit their questions, second a complementary mechanism for users to deliver answers to questions, and third a web-based platform to facilitate user interactions. Online forums have acted as a CQA service function ever since the beginning of the Internet — so in that sense CQA is nothing new. Websites devoted to CQA, however, appeared only in recent years; the first CQA service, the Korean Naver Knowledge iN, was launched in 2002. The first English CQA site, Answerbag, was not launched until April 2003. CQA services have proliferated in the past eight years or so (if we consider the launching of Yahoo! Answers in 2005 as the milestone), as a rising market for the fulfillment of various user intents. It has been reported that the number of questions answered in CQA services by far surpasses the number of questions answered by library reference services [70], which used to be the major platform for such question answering (questions there were mostly answered by a specific individual). In October 2009, Yahoo! Answers had over 200 million users, from which there are more than 1.5 million users visits the site on the daily basis. By May 2010, it has provided more than one billion questions, with on average one question generated in every 10 seconds; the number of questions submitted to the Chinese CQA service Baidu Knows, so far, has surpassed 155 million, with a daily volume of 10 million user visit.

Yahoo! Answers

Figure 2.1: An example of Yahoo! Answers question

Yahoo! Answers is arguably the most successful CQA service (a typical example of Yahoo! Answers can be found in Figure 2.1), with the largest CQA population in the English language.

The general idea behind the design of Yahoo! Answers is to strike the optimal balance between having a large number of users and having a high quality of answers. As shown in Figure 2.2, Yahoo! Answers set up a time frame for each question to obtain answers from community, which enables the service to remove the low quality questions — most users do not enjoy answering noninformative questions. Specifically speaking, once the asker submits the question, it will remain open for four days awaiting good answer candidates from the public. Once two or more answers are collected, the asker can either pick the Best Answer or leave it for the community to decide the Best Answer by vote (if the asker did not pick the Best Answer). Notice that when a question receives only one answer, the user can extend the open period for another 4 days to allow more possible answers to be generated. If the asker's question still cannot resolved after 8 days, it

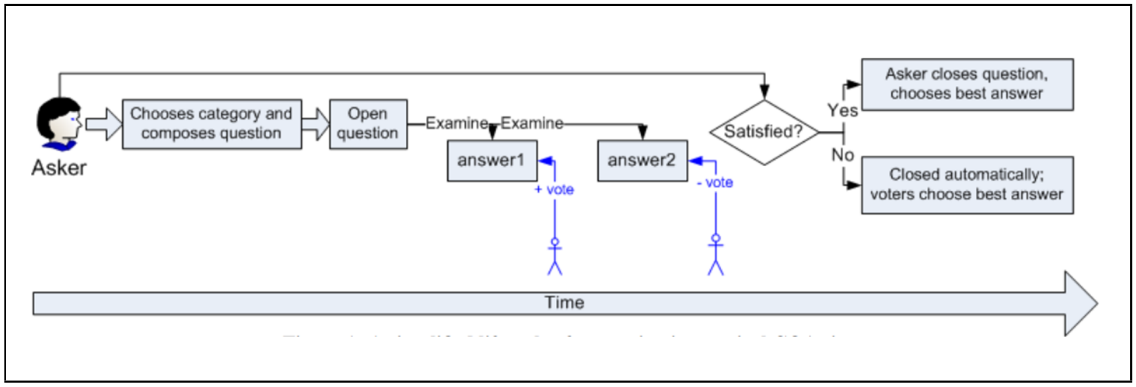

Figure 2.2: The simplified lifecycle of a question in Yahoo! Answers

automatically goes to a vote with the option of either the "Best Answer" or "No Best Answer". If the "No Best Answer Option" wins the vote then the question would be automatically removed and the asker redeems five points back to his/her account. If a question does not receive any answers within 4 days, it would be regarded as spam and is deleted. But unlike other CQA services like Quora, in which users are allowed to use points as tokens to invite experts of the community to answer a new question (this would force the asker to paraphrase the question to prevent the credit loss), points in Yahoo! Answers are rather the indicator of his/her community status with certain operational privileges. For example, first level users can answer 30 questions each day while registered novice users are only allowed to ask and answer upto 20 questions on a daily basis.

Despite the success of Yahoo! Answers, it has been reported that Yahoo! Answers has a poor performance in resolving fact-driven questions [19]. This is because experienced users only make up with a small proportion at the user-base, and regular users usually have little interest in answering difficult questions.

WikiAnswers

WikiAnswers is a wiki-based website with web pages on various topics. It is similar to Yahoo! Answers in that users have to register with a username in

order to ask and answer a new question. The difference to Yahoo! Answers lies in the wiki technology, which allows communal ownership at the information. Each questions can have only one answer, which is continually edited and improved over time. The most active users are entitled to become volunteer supervisors, who are given the privilege to make certain high-level edits. With these privileges, they are encouraged to remove identical questions, delete vandalism questions, and transform conversational posts into answers.

In order to maintain the operation of the service, there are two types of volunteer Supervisors in WikiAnswers namely, Category and Floating. Category Supervisors are obligated to manage one or more categories in which he/she excels. Usually, people who possess some unique expertise will be requested to become one of the Category Supervisors. Floating Supervisors can access the same privilege as Category Supervisors, but with no restriction of some certain categories of the questions. It is more flexible for those who only have a limited time to get started. There are also Senior Supervisors, who are selected from experienced supervisors (both the Category and Floating). Senior Supervisors are responsible for guarding the Top Categories on the site, and assisting new supervisors when their mentors are not available online. They may help out with some minor disagreements but the conflicts of the sites are usually resolved by a dispute resolution process, in which paid staff called Community Assistants will make the final judgment. The site also has a group of Advanced Supervisors. These supervisors are normally selected from the best Senior Supervisors and are deemed as the most privilege supervisors with the power to give a final verdict.

In light of the success of Yahoo! Answers and Wiki Answers, a great deal of CQA services, with users' social media identity, have emerged and become popular. By integrating social media ingredients, those CQA services can help users to obtain information in a more collaborative fashion — a user can forward his or

her questions to his friend's circle, which allows the questions to be solved by the power of friends-of-friends, since some of them may be familiar with asker's background and some of the others may share a common interest. Typical examples include Quora[1], Facebook Questions[2], and also domain-specific forums like Stack Overflow[3].

eHow

eHow is a how-to guide which consists of more than 1 million articles, supplying users with step-by-step instructions. eHow articles cover a wide range of topics which are comparable to Yahoo! Answers, and the article writers are usually freelancers who get paid by the quality and amount of articles. Any eHow user can give comments to the article answer, but only the article writers have the privilege to change the content of the articles.

Facebook Questions

In May 2010, Facebook published Questions, with the aim to compete with the Yahoo! Answers service. In addition to the features like communal ownership and real identity, Facebook Questions provides a recommendation links (according to the question types and topics) which steer users to relevant items in Facebook's repository of "fan pages." This feature helps to pinpoint user intent when looking for items such as movie recommendations or restaurant reviews.

Quora

Quora was originally followed by experienced internet users, such as internet entrepreneurs and software geeks, who are at the heart of the platform. The quality of the questions submitted here is remarkably better than other CQA services, such as Yahoo! Answers, for two reasons. First, expert users are at the core of Quora, whose question is then distributed to other regular users and get endorsed by other

[1] http://www.quora.com/
[2] http://www.facebook.com/questions
[3] http://stackoverflow.com/

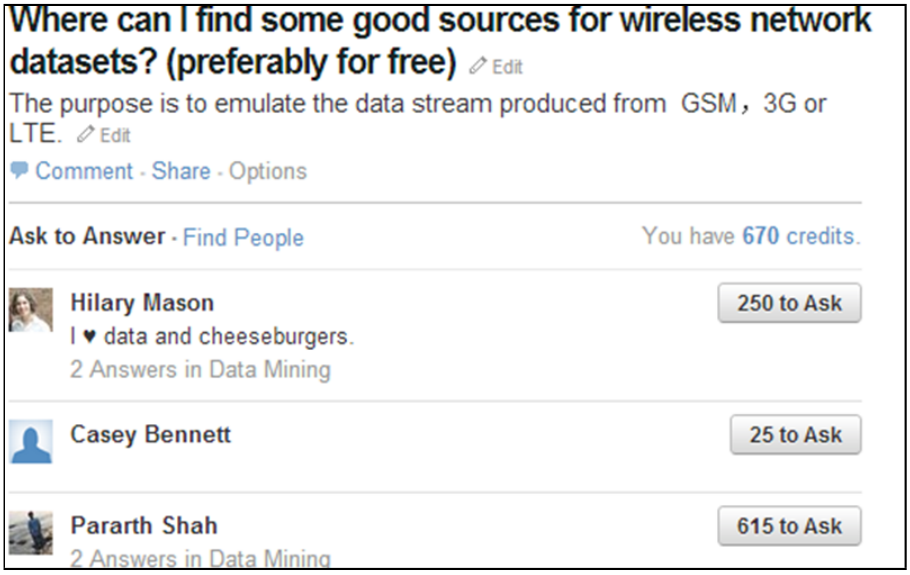

Figure 2.3: an illustration of how Quora assists user to find experts to answer a question

regular users. Another reason is that the askers are expected to use their real identity when answering questions, whereas other CQA services usually require users to register a username. However, at the price of quality control, the population in Quora is much smaller than that of Yahoo! Answers and users are not as active as their counterpart in Yahoo! Answers. This is quite understandable: despite its success in distributing high-quality information (both questions and answers) to regular users, regular users in Quora may find their questions (or answers) can hardly draw attention from other regular users due to the quality control mechanism. They do not feel like they are the owners of the service but rather seekers or receivers of information.

Aardvark

Google also has its own question answering service namely Aardvark, which is designed with a similar rational as Facebook Questions. Users submitted questions via the Aardvark website, email or instant messenger and Aardvark identified and

facilitated a live chat or email conversation with the corresponding topic experts in the asker's extended social network. Aardvark was used for asking subjective questions for which human judgment or recommendation was desired. The Aardvark team was mostly moved to Google+, and that's probably due to the better use of Google resources.

StackOverFlow

StackOverFlow focuses on a wide range of topics in computer programming. Similar to the mechanism of Quora, users of StackOverflow can earn points and badges. If a user needs to resolve a difficult question, he/she can pay reputation points to other users as tokens (which are known as "bounty"). Users on StackOverflow are mostly technology geeks, who are often driven by the motives of winning the game and gaining reputation points.

2.2 Question Classification

In the traditional TREC QA track, question classification is arguably the most important component since it can help the QA system to understand the question type. Question classification is also an essential component in CQA which enables it to understand the question intent, it also allows other applications (such as question retrieval and answer validation) to exploit the inherent CQA category information. However, unlike web pages or documents, questions in CQA are usually quite short – the average question length in Yahoo! Answers is 9.92, not including the description part – Figure 2.1 shows a common example. The fundamental challenge is that questions in CQA do not have enough co-occurrences for the similarity calculation, so that the performance of the standard "bag of words" models is often very low due to the data sparseness.

Approaches to tackle this problem can be divided into two directions. The first direction is that of text representation enrichment by analysing the original

textual or metadata features with the purpose of discovering new patterns through corpus exploration. Approaches in this direction can be traced back to the era of factoid question answering. Various systems were developed, but the basic idea is the same: classifying the questions into predefined categories and recognizing the corresponding entities in the relevant documents. For example, in the traditional TREC QA classification, Li et al. [44] presented a two-stage question taxonomy which comprises six top-level coarse-grained classes, such as location and numerics, and fifty bottom-level fine-grained classes, such as city and country. They developed a hierarchical classifier which classifies questions into fine-grained classes, according to their proposed semantic hierarchy of answer types. Approaches along these lines often require techniques for the extraction of syntactic features, from which the tree kernel approach is probably the most robust and effective one (it produces stable accuracy but does not require any human labelling process for the training dataset construction). For example, [87] proposed a special kernel function, known as the tree kernel which we mentioned above, to enable Support Vector Machines (SVM) to use the syntactic structures of questions. However, lexico-syntactic techniques (e.g., parsing) are not always viable here, since applying them to analyze the structure of the question texts is a time consuming process. Lin et al. [45] employed unigram and bigram words as features, for both the question and question description, under a hierarchical SVM classifier, and their results indicated that the introduction of question descriptions produce little improvement for the classification performance. Qu et al. [65] compared different learning models, namely Naive Bayes (NB), Maximum Entropy (ME), and SVM by assessing the classification performance on the Yahoo! Answers dataset. They conclude that hierarchical SVM with bag of words features overwhelms all the other models. Cai et al. [13] exploited the power of Yahoo! Answers categories to train the classifiers. They employed a search step to sift out the most relevant categories so as to allow the

classifier to concentrate only on a small closely related subset.

The second direction is to overcome the data sparsity by leveraging external resources, and often combining them with contextual information. Chen et al. [19] combined the texual features and metadata features so as to provide complementary insight of user intent. Jeong et al. [33] experimented with text representation enrichment which blends the use of syntactic-feature dependency and semantic-level WordNet hyponyms. However, WordNet cannot fully cover the colloquial language in CQA due to its limited vocabulary. Tu et al. [74] proposed a language modelling framework to expand documents with concepts (Wikipedia titles) as well as the relevant Wikipedia articles. However, the rich relations in Wikipedia, such as synonyms and associated terms, are simply discarded which leads to reduced performance. Wang et al. [80], later on, complement the previous model by incorporating enrichment relations from Wikipedia.

2.3 Question Retrieval

Question retrieval is another crucial component in a regular CQA service, which can resolve users' information needs straight away by helping the user to access the most similar questions. The first endeavor of question search can be traced back to the era of Frequently Asked Questions (FAQ) archives, which can be regarded as precursors to CQA archives, that attack similar problems but with a simpler interface, e.g, there are no features concerning users' profile, such as user experience and search preference. Jurczyk and Agichtein [2] reported a FAQ searching framework based on the Hyperlink-Induced Topic Search (HITS) algorithm, for the search task of a QA portal. They, later on, exploited interpersonal relationship to capture high-quality content, but they still did not answer the question of how to retrieve relevant questions.

One of the major challenges for question retrieval is the lexical gap between

the new question and the archived questions. Researchers have presented their interest in language modelling approaches for tackling this problem. Jeon et al. [31] designed a retrieval model based on translation models to identify similar questions from the the large scale archives, but the answer part was not exploited in their framework. Liu et al. [31] then proposed a similar approach with question-answer language model, which leverages the relationship within question-answer pairs for additional evidence. Cao et al. [14,15] examined the usefulness of question-category features for a category-based language model. Zhou [88] then reported that phrase-level features are usually more effective than features of word level. They argue that, in the translation probability learning process, contextual information should be considered as a whole rather than single words in isolation. Our framework in Chapter 8 is somewhat similar to the motivation of their work [15], but, unlike previous research which categorize each document as either topically relevant or irrelevant, our framework considers each archive document as a mixture of intents with a classifier output gauging the probability of each category. Moreover, previous works only incorporates textual features or category features alone, whereas in our work we also introduce a series of metadata features.

More recently, Ji et al. [35] developed their Question-Answer Topic Model (QATM) which leverages the facts that question and the corresponding answer usually share a similar topic. Instead of just investigating the question of a single sentence, Wang et al. [79] developed a multi-sentence questions retrieval model that focuses on questions with multiple sentences. They break down question into several components, which are topically related, from which the most appropriate fragments are selected to complement the original query. Later on they proposed another framework [78], particularly designed for retrieving questions of online language (questions without question mark or inquiry words).

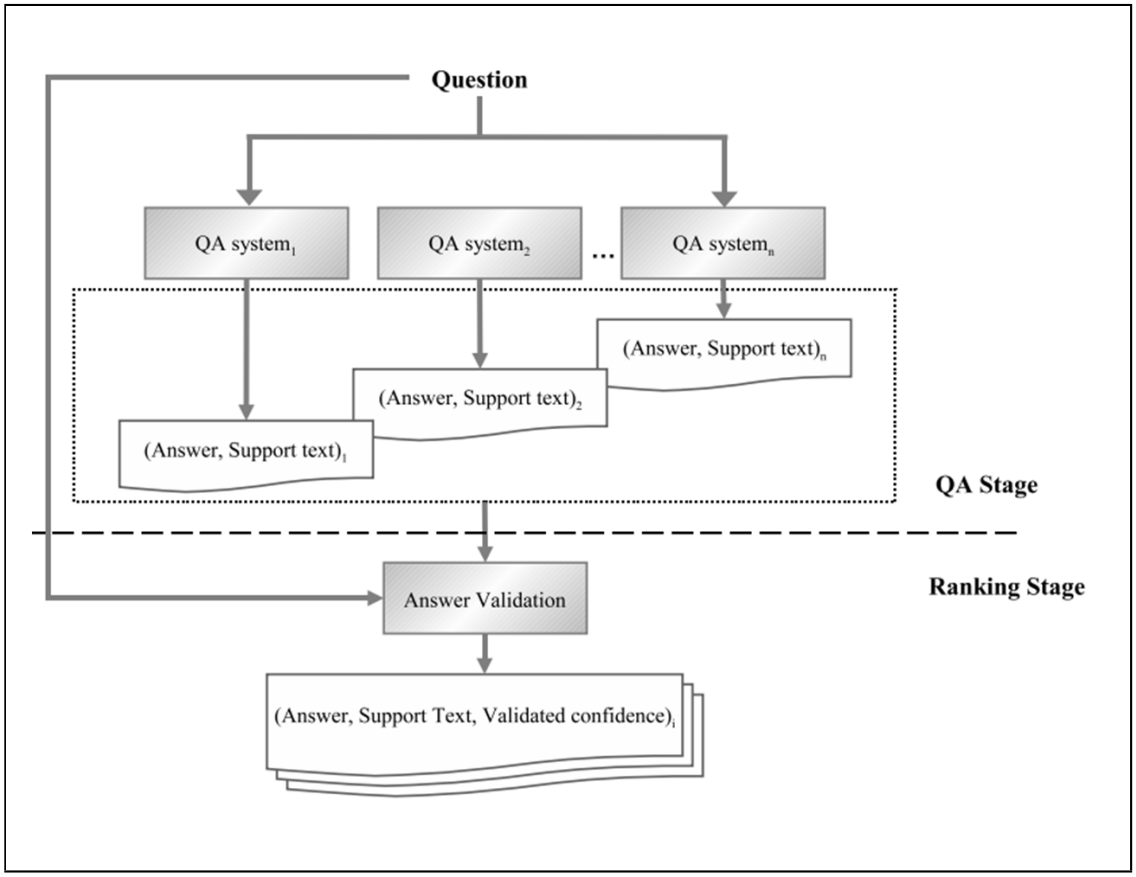

Figure 2.4: The flow chart of a typical answer validation system in CQA

2.4 Answer Validation

Answer validation endeavors to rank the candidates and assess to what extent the users' information needs can be satisfied. As shown in Figure 2.4, current automatic question answering systems are usually organized as a pipeline of reusable standard components for question analysis, answer generation, and answer validation, which is the final checkpoint regarding the answer quality. Even though answer validation has an immense potential to improve the performance of CQA services, unfortunately, most current CQA services do not incorporate answer validation since automatically answering questions is an extremely difficult task.

The most common way to validate the quality of an answer is that of measuring users' authority scores (a form of expertise). The rationale is that the question answerers are more likely to generate high-quality content than the question asker. For example, Jurczyk et al. [37] unravel several types of relationships intertwined in a community QA portal by modelling users' asking-answering, selecting best answers, and answer rating behaviors. Authority scores are calculated from users' asking-answering relationship, which is then incorporated in a regression model to predict answer quality. However, their work assumes that questions are all independent to each other. On the basis of their work, Suryanto et al. [73] then developed a more advanced model in which users' expertise are dependent, with an even better accuracy achieved. Perhaps the most fully developed set of evaluation criteria for answers is in the work of Zhu et al. [90] , where they identified and exploited a set of 13 criteria from both answer contents and the other comments provided by the community participants towards the questions and answers. Bian et al. [7] proposed a framework which is capable of measuring both answer quality and topical relevance. However, their work is still confined to the context of the factoid question answering with abundant labelled dataset available.

Another way of handling answer validation is by exploring the power of non-content or interpersonal features. Jeon et al. [32] proposed a model which makes use of the maximum entropy approach to estimate answer quality scores based on non-textual features. Their results showed that the most informative feature is the answer length, which is also confirmed by Agichtein et al. [2]. They introduced a general classification framework on the basis of the contributor relationships, which is then combined with textual and metadata features. They also conducted an in-depth investigation which reveals the 20 most informative features for the prediction of answer quality. Bian et al. [8] designed a semi-supervised framework, which is based on preference learning, to estimate the quality answers as well as

the corresponding users. Liu et al. [51] predicted answer quality by exploring the voting patterns from the users of a given topic. They considered the act that a user chooses the best answer as the indicator of information needs agreement. Based on this assumption, they identified users' satisfaction by making use of a graph model. Shah et al. [70] extend Liu's framework by considering answer rating as auxiliary evidence, by adding another constraint that the asker has to rate the chosen answer with at least 3 out of 5 stars. Shtok et al. [72] proposed a two stage model to measure users' satisfaction, from which users' satisfaction is captured by using features such as search list similarity and lexico-syntactics. Despite the success of these approaches, they mostly follow the stereotypes of casting the answer validation task into a classification problem, from which they learn the most informative features as the indicator for the answer quality.

From another perspective, it is also worth noting that Zobel et al. [91] first revealed that relevance judgement in search engine could be a subjective problem. It is, however, not until recently that researchers have started exploring the subjective relevance [51] in the context of answer validation, where a plethora of subjective, complex, and ill-formed contents are available for the exploitation.

2.5 Answer Recommendation

One of the most important issues of CQA services is that many appealing but challenging questions cannot be effectively resolved by answerers, it is therefore important to enable the user to have access to the members who are most likely to be able to answer the given questions.

Research on answer recommendation is largely related to another task, namely, group recommendation [59, 61]. Instead of recommending items (such as restaurants, markets, and websites) to a single user, group recommendation tasks aim to recommend items to a group of users. These two tasks are essentially analogical in

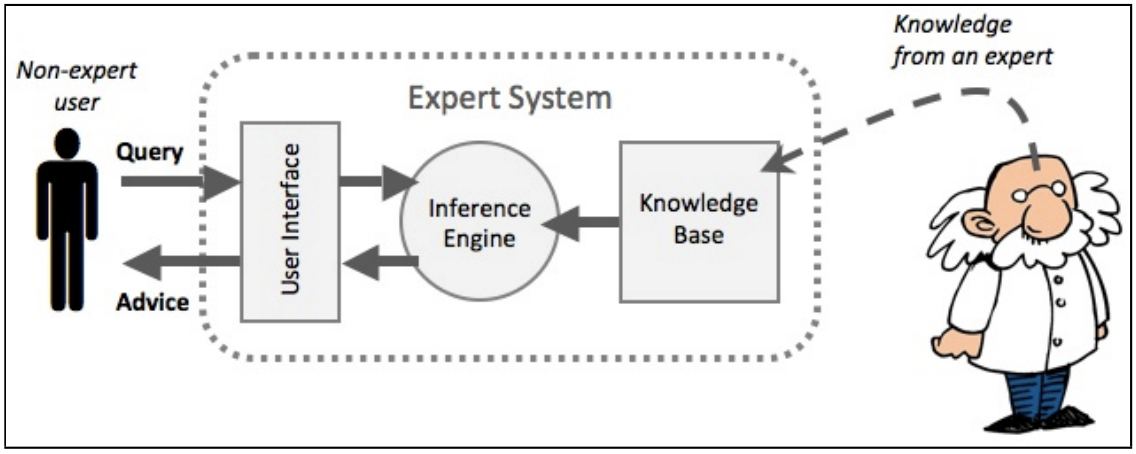

Figure 2.5: The flow chart of an expert system in CQA

that they both recommend items to various users. The key difference is that, the recommended items must be appealing to all group members for group recommendation, and so probabilistic aggregation approaches (such as EM Clustering) may be viable only for the group recommendation task. Furthermore, most of the CQA members are information seekers who do not have the habit of answering questions, so that the user-profile based on all members does not work with individuals.

An intuitive way to answer a new question is that of estimating user's expertise on the topic and forwarding the question to the domain experts, as shown in Figure 2.5. Approaches along these lines usually involve link analysis and latent topic modeling techniques. For example, Jurczyk et al. [36] proposed a graph model based on link analysis to calculate authoritative scores of users on the expected topics. Liu et al. [48] evaluate users' expertise by modelling answerer's interests in their searching history log, with a mixture of Language Model methods and Latent Dirichlet Allocation (LDA). Qu et al. [66] applied Probabilistic Latent Semantic Analysis (PLSA) to capture user interests on the basis of their answering history and interaction behaviors, from which they deduce the correlation between an answerer and a question. While approaches using PLSA are capable of identify-

ing whether users have the interest to answer a new question, they cannot answer the question to what extent these users expertise can match the questions with similar topical interest.

By capturing the structure of CQA, Riahi et al. [68] proposed a Segmented Topic Model (STM), which is more complex than LDA since it conducts a selection over high-level topics, to exploit more thematic features from the users' history. Bouguessa's model [10] considered users' authority level as a mixture of gamma distributions over each topic, which can automatically identify authoritative from non-authoritative users. Beyond the CQA context, there has also been similar research for online forums. For example, Ni et al. [58] designed a probabilistic generative model which is capable of learning potential topics for questions and users, and found that the best performance is attained when combining both the concept-level and word-level features for recommending a new answer.

Another attractive way for answer recommendation is to create more potential answerers to answer the question, which entails a deeper understanding of the user preference and interactive behaviors. The difference between these two approaches is that the former focuses on identifying the most likely experts for introducing high-quality and reliable answers, while the latter endeavors to explore more potential answerers who are capable of answering and contributing to the question. For example, Adamic et al. [1] investigated the use of the forum categories, and clusters them in terms of both textual features and patterns of user interaction. They concluded that a large proportion of Yahoo! Answers users tend to focus on contributing to domain-focused categories and the CQA service should recommend topic-wise questions to this group of users. Nam et al. [57] studied the motivation of top answerers, in which they summarise four types of answering motives, namely altruism, learning, competence and points. Users of each motive are more likely respond to the corresponding question with a similar incentive. Liu

et al. [49] explored the users' Web browsing history on the CQA systems, from which they designed a system which surveyed users' search preference. They report that search preference can have significant influence on answerers engagement skill, effort, and willingness to answer questions.

2.6 Research on User Intent

Another area closely related to this thesis is the study of user intent in Web search. The common paradigms of understanding user intent is by classifying the questions into several categories. This section will therefore quickly step through some major taxonomies of Web search, as well as some major techniques for measuring them.

In Broders seminal work [12], the users' intent is categorised as the informational, navigational and transactional. This is the most widely used taxonomy, and is considered as the basis for a variety of studies in the IR area. When one enters an informational query into search engine, he/she is looking for relevant information with the keywords. He/she is not looking for a specific site, as in a navigational query, and he/she is not looking to make a commercial transaction as with a transactional query. The user probably just wants to satisfy his/her information need. A navigational query is a search query entered with the intent of finding a particular website or webpage. For example, a user might enter "stackoverflow" into search bar to find the StackOverFlow site rather than typing the URL into a browser's navigation bar. A transactional search query is a query that indicates intent to complete a transaction, such as making a purchase. Transactional search queries may include exact brand and product names (like "iphone 5") or be generic (like "music player download") or actually include terms like "purchase," or "buy." In all of these examples, one can infer that the searcher is considering making a action in the near future. Baeza-Yates et al. [5] later on, proposed another taxonomy which classifies user intent as informational, not-informational or ambiguous. They

argue that a large proportion of user intent in Web search cannot be determined. Hu et al. [29] understood the user intent through the concepts introduced from Wikipedia; the authors summarised three specific intents: travel, personal name, and job finding. More recently, Jethava et al. [34] have studied the users intent as a multi-dimensional composition of different facets. They argued that the actual intent should not be decided only by one facet but by the correlations between them.

Existing methods for capturing user intent can generally be divided into two categories, namely Context-Aware methods and Context-Oblivious ones.

Context-Aware methods learn users intents according to search behaviors such as the current search query and associated URLs. Since queries are generally short with limited informativeness, it is natural for search paradigms to introduce query expansion for information enrichment. Kang et al. [38] combine document content, links, and URLs features to explore users navigational and transactional intent. Lee et al. [41] incorporates the past users' click-through behaviors and the anchor text distribution for the identification of navigational intent and informational intent.

Context-Oblivious methods are based on the assumption that adjacent user behaviors have the same or at least very similar user intents. Methods along this line concatenate users behavioral sequences as time series, from which user intent is then inferred. A number of advanced machine learning techniques have been applied to identify user intent in this direction, such as conditional random fields (CRF) [34] and sparse hidden-dynamics CRF models [71]. However, Context-Oblivious methods have only achieved a limited success due to the reduced feature space, which is also confirmed in [71] which reported that Context-Aware approaches generally outperform Context-Oblivious ones.

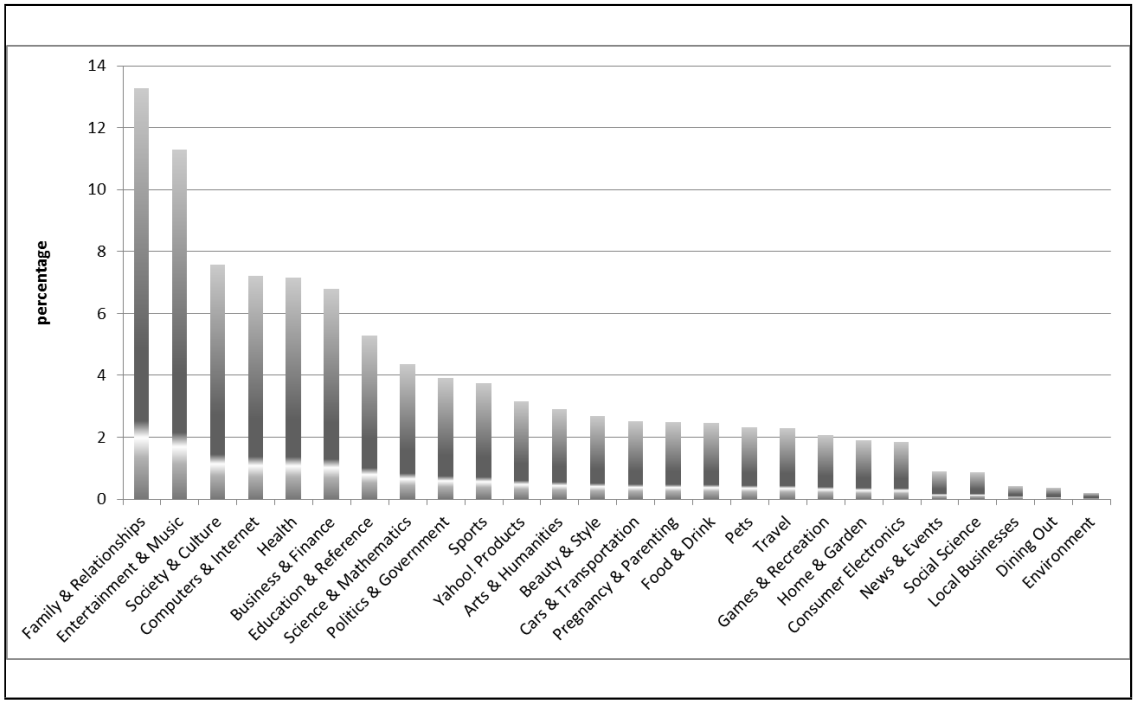

Figure 2.6: The question distribution over Yahoo! Answers categories

2.7 Datasets Used in the Thesis

Yahoo! Answers Dataset Experimental data is derived from a subset of the Yahoo! Research Alliance Webscope[4] program - Yahoo! Answers Comprehensive Questions and Answers, version 1.0, which has been made public to all interested researchers. We choose it as our experimental dataset because it is the only one that has been authorised with a large amount of meta-data information (users' private information has been made anonymous). The original corpus consists of 4,483,032 questions and their corresponding answers from 2005/01/01 to 2006/01/01.

Figure 2.6 depicts the question distribution over the 26 Yahoo! Answers top Categories. It is notable that hot topics in Yahoo! Answers includes *Family & Relationships, Entertainment & Music, and Society & Culture*, which confirms

[4] http://webscope.sandbox.yahoo.com

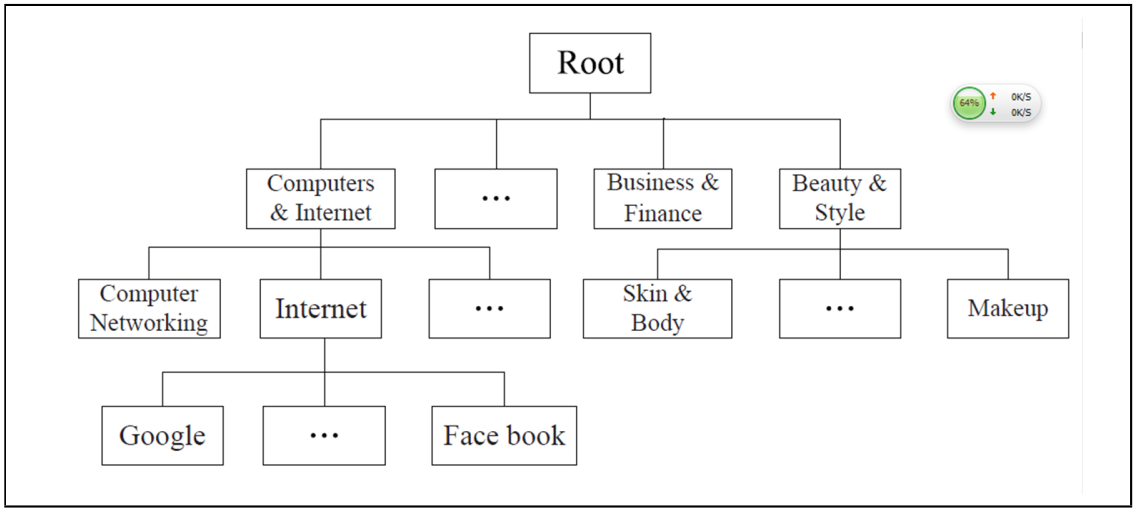

Figure 2.7: The taxonomy of Yahoo! Answers

the subjective and decentralised nature of Yahoo! Answers we have speculated in Section 1.3. The least active categories are *Environment* and *Dining Out* which suggests that task-driven questions are not very attractive topics among the Yahoo! Answers users.

Figure 2.7 depicts the top level of Yahoo! Answers taxonomy. The hierarchy taxonomy supplies users means of managing data at different levels of abstraction. In Figure 2.7, each node (or category) corresponds to a topic which is composed by a group of questions. An edge between two nodes represents the supertype-subtype relation. It has been reported in [14, 85] that the hierarchical category structure can be used to improve the performance of question retrieval, and thus assisting in the understanding of user intent.

In addition, there are 2,665,298 askers overall with each of them on average submitting 1.682 questions. There are 621,349 best answerers, with each of them on average answering 7.215 questions. It is clear that Yahoo! Answers suffers from the answerer starvation problem — the phenomenon that people enjoy asking instead of answering questions, which results in a large number of unanswered questions.

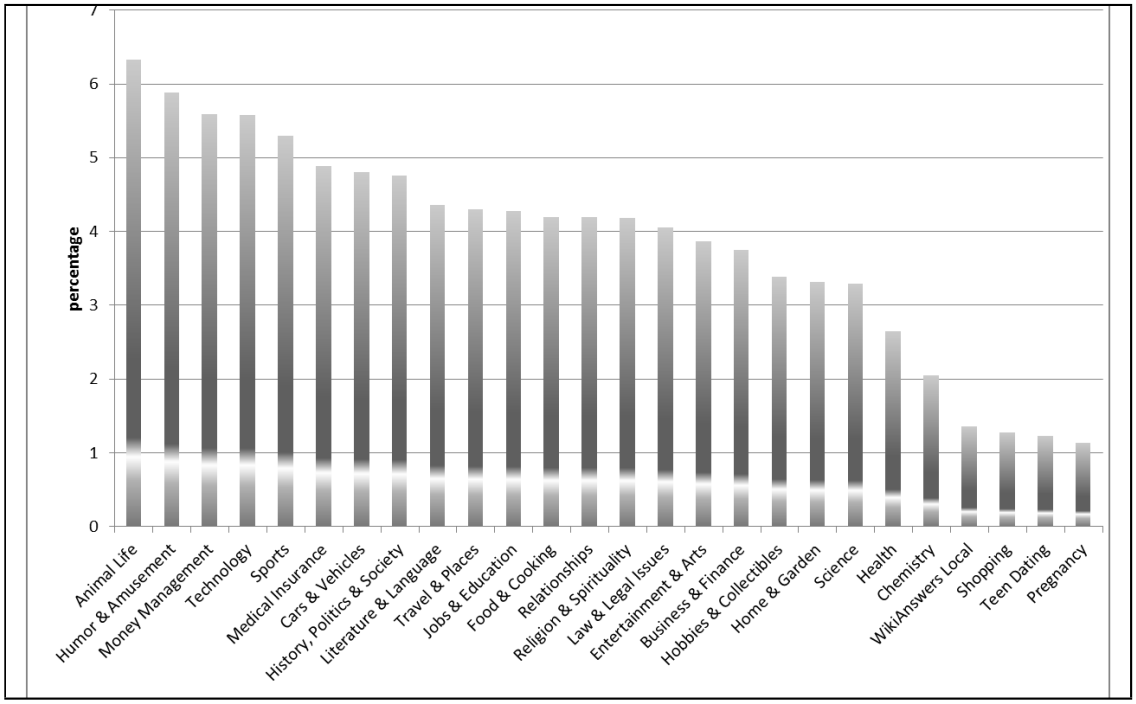

Figure 2.8: The question distribution over WikiAnswers categories

For example, it has been reported in [72] that around 15% of all incoming English questions ended up having no answers and were deleted in Yahoo! Answers in 2011.

WikiAnswers Dataset

The WikiAnswers dataset was collected by us from WikiAnswers[5], dating from 2012/01/01 to 2012/05/01 contains a total of 824,320 questions (note that this is only a subset and cannot cover all the questions during that period of time). All the local questions are derived from the WikiAnswers Local category as we find this is the only category in WikiAnswers that is completely devoted to locality intent. We will present the detailed statistics regarding the test and training sets, and validation set in Table 4.1.

Figure 2.8 reports the questions distribution over the WikiAnswers categories, from which we can see the most popular categories are *Animal Life, Humor*

[5] http://wiki.answers.com/

& Amusement, and *Money Management*. *Animal Life* and *Money Management* categories are information-driven topics while *Humor & Amusement* is a subjective one. This phenomenon indicates that, unlike Yahoo! Answers users who are asking questions with a heavily subjective and social style, WikiAnswers users tend to ask questions in a more balanced manner.

Chapter 3

Understanding Users' Objective/Subjective/Social Intent

One of the most active areas of research in NLP is arguably the extraction of opinions and emotions from the text, which can be employed as an auxiliary tool to improve the performance of applications such as Web search, information extraction, and question answering. This chapter focuses on understanding user's objective/subjective/social (OSS) intent through a semi-supervised learning approach. The rest of this chapter is organised as follows. In Section 3.1, we introduce the background of OSS intent in CQA. In Section 3.2, we review the related work regarding OSS intent in CQA. In Section 3.3, we give detailed definitions of users' OSS intent. In Section 3.4, we investigate the usefulness of text and metadata features for identifying the user intent of questions, and also present the Co-Training approach to question classification. In Section 3.5, we describe the experimental setup and present the experimental results. In Section 3.6, we conclude this chapter.

3.1 Overview of OSS Intent

Opinion Mining is the task of identifying the viewpoint of the text. Summarizing these viewpoints have a good potential in helping many business and organizations, where the sentiment of the customer on a product is needed, or an individual wants to know other people's opinion. Most techniques along these lines rely on the lexical/syntactical of text. However, words may have both subjective and objective senses, which is a source of ambiguity in opinion mining. For example, it has been reported in [52] that even the words, which are proved as reliable clues of objectivity, may have non-negligible degrees of subjective sense.

Although questions submitted to CQA services are typically seeking *objective* knowledge, there do exist many questions that ask *subjective* opinion or *social* interactions: First, the factual knowledge available in CQA cannot satisfy all users' information needs, since very often a question does not have a single definitely correct answer, but the asker is interested in what others' thoughts are. Second, many askers go to CQA services simply with the aim to build up online social engagement (no matter how loosely it is) rather than resolving certain information needs. A promising way for CQA services to handle this problem is to classify the user intent into several types and treat each type in a different fashion (given a reasonably high accuracy of classification), e.g. in question retrieval (see Chapter 8, Section 8.4). In this chapter, we describe the identification of OSS intent in CQA. Specifically, in order to identify the user intent of a new question, we build a predictive model through machine learning based on both text and metadata features. Our investigation reveals that these two types of features are conditionally independent, and each of them is sufficient for prediction, therefore they can be exploited as two views in Co-Training [9] — a semi-supervised learning framework — to make use of a large amount of unlabelled questions, in addition to the small set of manually labelled questions for enhanced question classification. The preliminary

experimental results show that Co-Training works significantly better than simply pooling these two types of features together.

3.2 Previous Work on OSS Intent

The objective/subjective intent of questions has been investigated by researchers before. For example, in TREC competitions, subjective/complex question answering were initially addressed in the opinion QA track from 2007 [20]. The work most similar to ours is [43] in which Li et al. use supervised and semi-supervised machine learning methods to predict the subjectivity orientation of questions, i.e., whether a user is seeking objective or subjective information. However, their proposed approach relies on features extracted from both questions and their corresponding answers, therefore it can only be used to classify questions that have already been answered. In contrast, our approach aims to classify questions instantly once they are asked so only features extracted from questions are used. Thus a CQA system can identify a new question's underlying user intent through our approach and furthermore exploit it to improve the question answering process (e.g., in finding similar questions or relevant answers).

The social content of questions has also received some attention from researchers recently. Liu et al. have extended Broder's taxonomy of Web search queries to include a social category for CQA questions [52]. However, as mentioned above, that taxonomy is not really suitable for CQA. For example, the navigational category in their study literally contains no questions at all. Furthermore, as mentioned in Section 2.6, Rodrigues and Milic-Frayling have analysed the social vs. non-social intent of questions in CQA [55], but their definition for social intent is quite different from ours, as they mainly focus on defining measures of social engagement to characterise users' participation and contribution. Harper et al. have proposed to describe the user intent of questions in CQA as informational

and conversational [27]. Their conversational category is somewhat similar to our social category.

3.3 Research Problem Pertaining to OSS Intent

Taking into account the special characteristics of CQA, we propose the following taxonomy that classifies questions into three categories of user intent according to what type of answers they seek: *objective*, *subjective*, and *social*. Thus we formulate the user intent understanding problem as a question classification problem.

Objective Questions The intent of such questions is to get factual knowledge about something. For example, in Yahoo! Answers, the question "Which country in Africa that was colonized by France did assimilation policy succeed?" asks for specific details of a particular event. As another example, the question "How do I find the website for the brick township high school baseball team for this year 2006?" asks for the website address where the user can learn more details about a particular entity.

Subjective Questions The intent of such questions is to get personal opinions or general advice about something. For example, the question "Do you believe Canada's flag should be lowered for each soldier that dies in the service of their country?" asks for personal opinions about a topic which could be very different for different people due to different upbringing and background. As another example, the question "I am a Bangladeshi National girl and I came to USA on B1/B2 visa and now I would like to take admission pls adv?" asks for general advice on a complicated issue.

Social Questions The intent of such questions is not to seek information but to have social interactions with other users. For example, the question "i am 4m

kolkata,india.any1 4m here want to be my frnd?gals or guys- no prob with that.betr if a teenagr.i'm 17" and the question "Any1 near Newyork city?" are trying to make friends. For another example, the question "why do people from the USA ask questions as if that is the only country on the web?" is probably trying to get some empathy from people with similar thoughts.

The objective category in the above taxonomy refers to the traditional TREC-style questions, while incorporating both the subjective category and the social category simultaneously distinguishes it from existing taxonomies for CQA questions which only focus on one of them.

Most questions that we encounter in a CQA service can be classified into one of these three categories. However, it is possible to see ambiguous questions. For example, the question "What type of careers are in southeast asia?" could either be interpreted as objective (asking for career facts) or subjective (asking for career advice). After careful inspection of the dataset, we observe that such questions constitute less than 2% of all questions, so we ignore them in this thesis.

Although examining the answers to a question usually helps to infer its user intent accurately, we prefer to utilise the question alone because only by predicting the user intent of a question before it receives answers, could we exploit the user intent to enhance the question answering process in CQA.

3.4 Approach to Dealing with OSS Intent

To shed light on users' objective, subjective, and social intent in CQA, we present a method for automatically assigning labels in the taxonomy to questions, which uses metadata features and integrates more diverse types of knowledge than in previous work.

3.4.1 Textual Features

The textual features of a question are extracted from the bag-of-words content of the question title after standard pre-processing steps (tokenization, lower-casing, stopword-removal, and stemming) [53]. Finally each question is represented as a vector of unigram and bigrams words weighted by TF×IDF [53].

Now the first step for understanding the *information gain* is deciding what features of the data are relevant to target class we want to predict. We can build a decision tree in a top-down fashion, but the question is how to choose which attribute to split at each node? The answer is find the feature that best splits the target class into the purest possible children nodes (which are the nodes that don't contain a mix of both classes, rather pure nodes with only one class). This measure of purity is called the information. It represents the expected amount of information that would be needed to specify whether a new instance should be classified in which class. We calculate it based on the number of each classes at the node. Entropy on the other hand is a measure of impurity (the opposite). It is defined (for a binary class with values a/b) as:

$$H(T) = -p(a) * log(p(a)) - p(b) * log(p(b)) \tag{3.1}$$

$$IG(T, k) = H(T) - H(T|k) \tag{3.2}$$

where H denotes the entropy, T denotes the training examples, k is a random attribute in an example. Therefore, according to the definition: $H(T)$ represents the amount of entropy before the split, $H(T|k)$ represents the amount of entropy after the split. The information gain is equal to how much information we gained by doing the split using a particular feature.

To have a rough idea about each category of questions, we sort all unigram and bigram word features in terms of information gain (3.2) for question classification, and show the most discriminative ones in Table 3.1.

It seems that questions with those 5-w words (who, when, where, what, why) are more likely to have an objective intent, whereas questions with polite words and conversational phrases are more likely to have a subjective or social intent. This suggests that textual features have relatively more discriminative power for identifying objective questions than separating subjective and social questions.

3.4.2 Metadata Features

Moreover, we have also identified several metadata features that can work in addition to textual features.

Question Topic

Figure 3.1 shows the distribution of user intent over the top-10 question topic categories (all questions posted in Yahoo! Answers are annotated by their topic categories). It seems that objective and subjective questions have a similar proportion of presence in most topic categories, except for "Arts & Humanities" which contains many subjective questions about history and genealogy. The distribution of social questions seems to be quite different from the other two kinds of user intent: most social questions are about topics like "Family Relationships", "News Events", and "Entertainment & Music" on which people may be more inclined to anticipate social interaction. This suggests that question topic features have relatively more discriminative power to separate social questions from objective or subjective questions.

Question Time

Figure 3.2 shows the distribution of user intent over the time (hour-of-the-day) when the question was asked on 1st May 2006. It seems that objective and

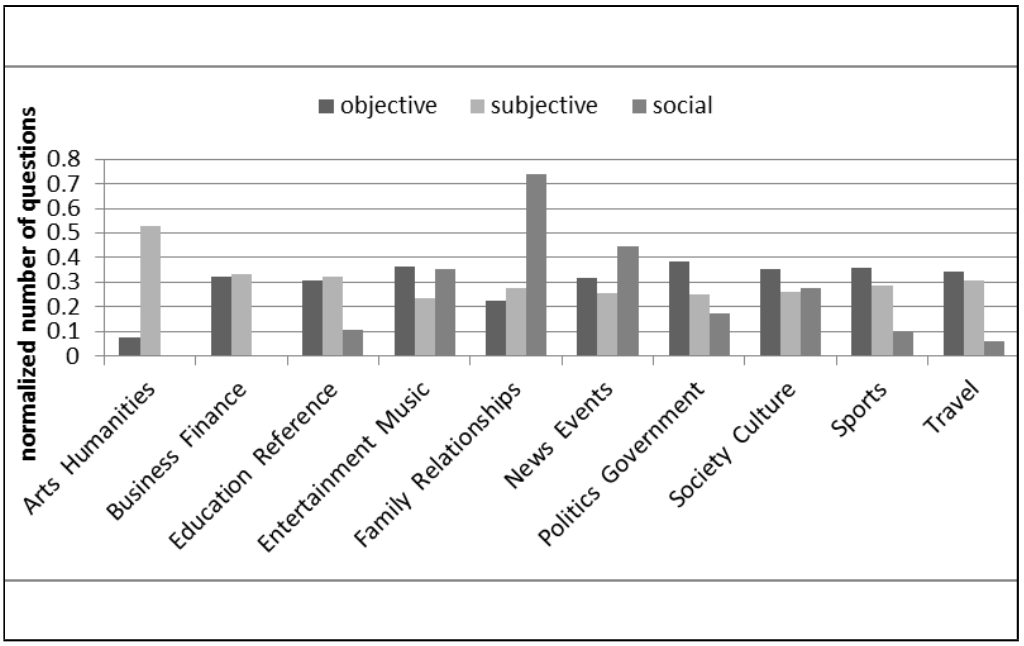

Figure 3.1: The question topic feature

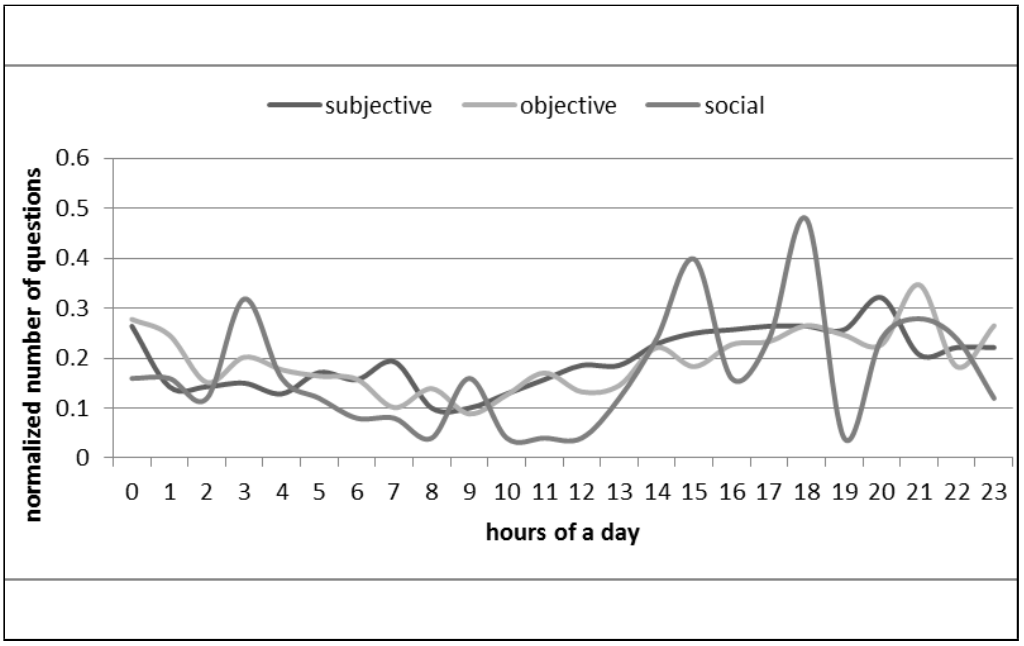

Figure 3.2: The question time feature

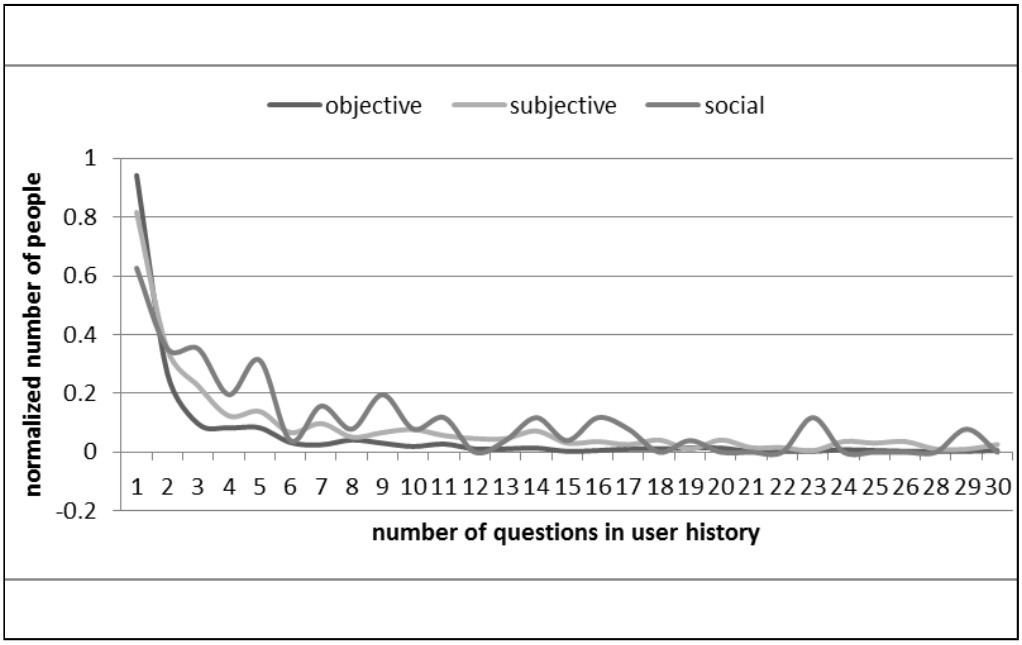

Figure 3.3: The question asking experience feature

subjective questions do not have apparent differences in terms of question time. In contrast, social questions show interesting patterns: the peak time for social questions is at 18:00 (finishing the day-time work), 15:00 (after lunch), and 03:00 (lonely in the late night). This suggests that question time features have relatively more discriminative power to separate social questions from objective or subjective questions.

Question Asker's Experience

Figure 3.3 shows the distribution of user intent over the question asker's experience measured by the number of questions that the user has asked before. It seems that subjective and social questions are more likely to come from experienced users than new users, probably because experienced users recognise that the main strength of CQA is in subjective or social questions but not objective questions, compared with Web search engines. This suggests that question asker experience features have relatively more discriminative power to separate objective questions

from subjective or social questions.

3.4.3 Co-Training

It is time-consuming and error-prone to manually label questions according to their user intent. Usually we can only have a small set of labelled questions, which would seriously limit the success of supervised learning for question classification. However, obtaining unlabelled questions is quite easy and cheap. So it is promising to apply semi-supervised learning [18], which can make use of a large amount of unlabelled data in addition to the small set of labelled data. By doing so, CQA services can minimise the chance of wrongly guessing the user intent, which may lead to the a scenario of assigning low-quality answers to a new question.

There are many semi-supervised learning techniques available. For this problem of question classification according to user intent, we believe that the Co-Training [9] approach is particularly suitable. Basically, Co-Training is a semi-supervised learning framework that requires two views of the data: each example is described by two different feature sets (views) that provide different, complementary information. In the ideal situation, the two views are conditionally independent (given the class) and each view is sufficient (to be used for classification on its own). The main steps of Co-Training are as follows. It first learns a separate classifier for each view from the labelled data, and then the most confident predictions of each classifier on the unlabelled data are used to construct additional labelled training examples. This process is iterated until a stopping criterion is met.

As we have pointed out in Section 3.4, the text and metadata features are both effective in detecting the user intent of questions but with quite different discriminative powers for different question categories. Therefore they can be considered as the two views for Co-Training.

Our implementation of Co-Training framework is similar to that of [9], which

is described in Algorithm 1.

Algorithm 1: Co-Training

Data: Input data, which includes:

F_{text} and F_{meta} which represent textual features and metadata features respectively

C_{text} and C_{meta} which are classifiers based on F_{text} and F_{meta} respectively

L_{train} that is the set of labelled training examples

L_{test} that is the set of labelled testing examples

U is the unlabelled training examples

N is the maximum iteration limit

Result: The performance of the last iteration

```
while i < N do
    repeat
        Use C_text to classify all the examples in U based on F_text ;
        Select the top K_Q examples with the highest confidence of
        prediction;
        Remove those examples from U and add them to L_train;
        Using C_meta to classify all examples from U based on F_meta;
        Select the top K_H examples with the highest confidence of
        prediction;
        Remove those examples from U and add them to L_train;
    until there are no unlabelled examples left in U;
end
```

3.5 Experiments on OSS Intent

3.5.1 Dataset

Table 3.2 shows the statistics about the dataset for experiments. It consists of 1,539 questions that were randomly selected from the original Yahoo! Answers dataset and manually labelled according to their user intent. Three people (a woman and two men) were asked to label the dataset, who assigned a label (as either objective, subjective, or social) to every question. We considered those labels, which two-third of the voters agree on, as final labels of the questions. The dataset is then split randomly into training and testing sets with a proportion of 2:1.

3.5.2 Performance Measure

Since the class sizes are imbalanced in this problem, we use the F_1 score [53] instead of accuracy to measure the performance of question classification. The F_1 score is the harmonic mean of precision P and recall R: $F_1 = \frac{2PR}{P+R}$, where $P = \frac{\text{true positive}}{\text{true positive + false positive}}$, $R = \frac{\text{true positive}}{\text{true positive + false negative}}$. Furthermore, both micro-averaged F_1 (miF_1) and macro-averaged F_1 (maF_1) [83] will be reported in the next section. The former carries out averaging over all test questions while the latter over all question categories, therefore the former is dominated by performance on major question categories while the latter treats all question categories equally.

3.5.3 Results

A number of machine learning algorithms implemented in Weka[1] , including C4.5, Random Forest, Naive Bayes, k-Nearest-Neighbours, and Support Vector Machine (SVM) , have been tried out for both supervised learning and semi-supervised

[1] http://www.cs.waikato.ac.nz/ml/weka/

learning (Co-Training). SVM has delivered the best classification performance in our experiments, so we only has its results here.

3.5.3.1 Supervised Learning

Table 3.3 shows the performance (miF_1) of question classification through supervised learning with different sets of features. The Linear SVM parameters are set to their default values except that the class weights are optimised for each question category by 5-fold cross-validation.

It is obvious that using both textual features and metadata features works better than using either kind of features alone, for all question categories.

The performance improvement brought by using metadata features in addition to textual features for supervised learning is statistically significant ($P < 0.025$), according to the micro sign test (s-test) [83].

3.5.3.2 Semi-Supervised Learning

Table 3.4 shows the performances (miF_1 and maF_1) of question classification through supervised learning and also semi-supervised learning (Co-Training) based on both text and metadata features. The Linear SVM parameters are set as in supervised learning, while the Co-Training algorithm parameters are tuned to their optimal values via 5-fold cross-validation.

It is obvious that the Co-Training approach that regards textual features and metadata features as two views works better than the supervised learning approach that simply pooling these two types of features together. This is probably because Co-Training, as a semi-supervised learning method, can make use of a large amount of unlabelled questions in addition to the small set of labelled questions.

The performance improvement brought by using unlabelled data in addition to labelled data through Co-Training rather than simply combining text and

metadata features together is statistically significant ($P < 0.005$), according to the micro sign test (s-test) [83].

Figure 3.4 shows the performance of Co-Training over iterations with the optimal incremental size. For miF_1, the optimal performance is achieved at the 13th iteration (with 260 unlabelled questions being added to the training set each round). For maF_1, the optimal performance is achieved at the 25th iteration (with 150 unlabelled questions being added to the training set each round). Choosing a smaller incremental size could lead to a better performance, but meanwhile it would require more iterations and thus be less efficient.

Figure 3.5 shows the performance of Co-Training vs supervised learning with varying number of labelled questions available. It can be seen that Co-Training consistently outperforms supervised learning with a substantial gap for miF_1, though there is no clear winner for maF_1. Furthermore, Co-Training only needs about 30% of labelled questions to reach the same miF_1 performance as supervised learning.

3.6 Summary

The main contribution of this chapter is threefold. First, we propose a taxonomy of user intent in CQA that incorporates both the subjective/objective and informational/social perspectives. Second, we identify several metadata features which can be used together with standard textual features by machine learning algorithms to classify questions according to their underlying user intent. Third, we demonstrate that it is better to exploit both textual features and metadata features through the semi-supervise learning framework, Co-Training, rather than simply combining them in supervised learning, since the former can make use of a large amount of unlabelled data.

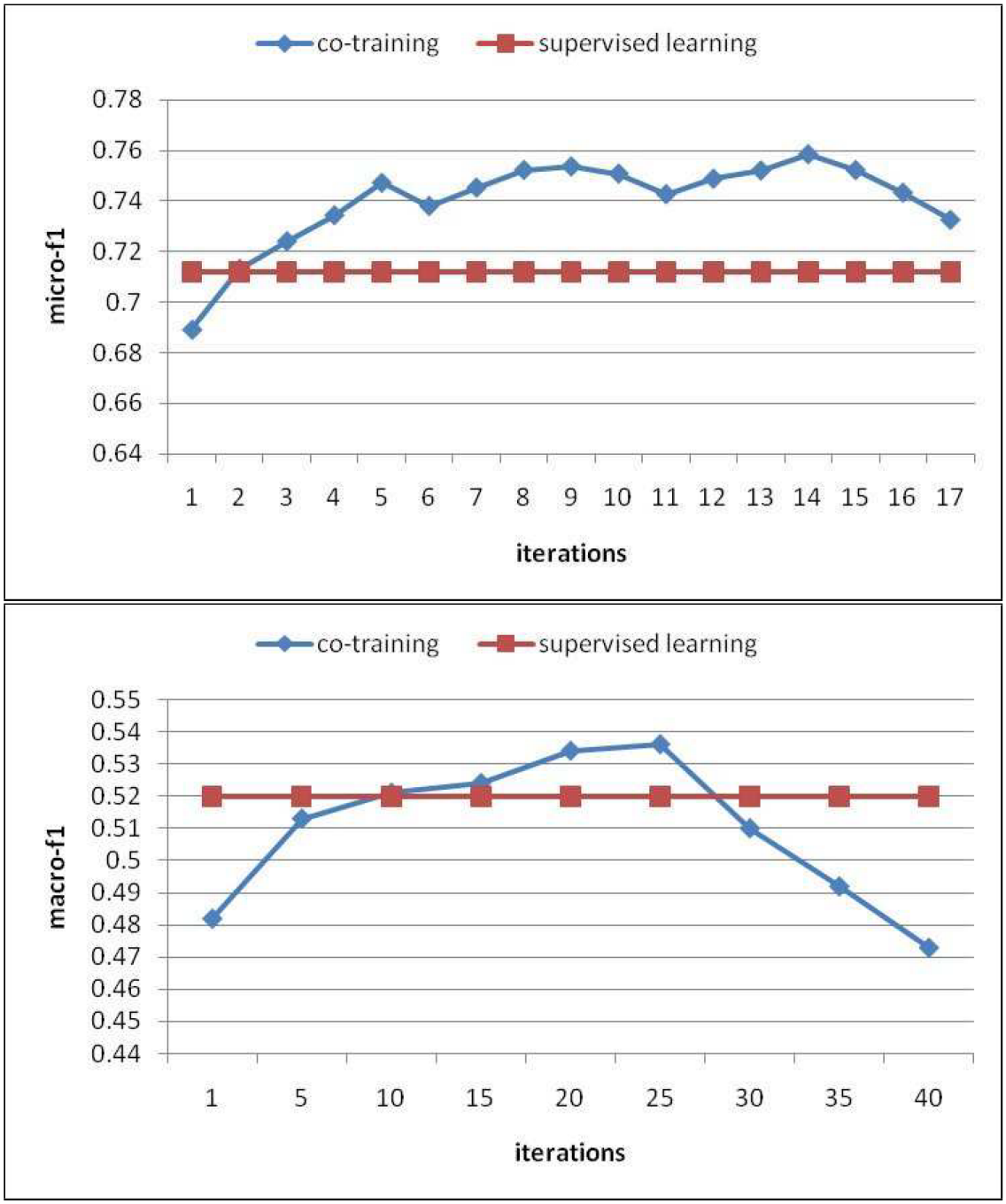

Figure 3.4: The performance of Co-Training over iterations with the optimal incremental size.

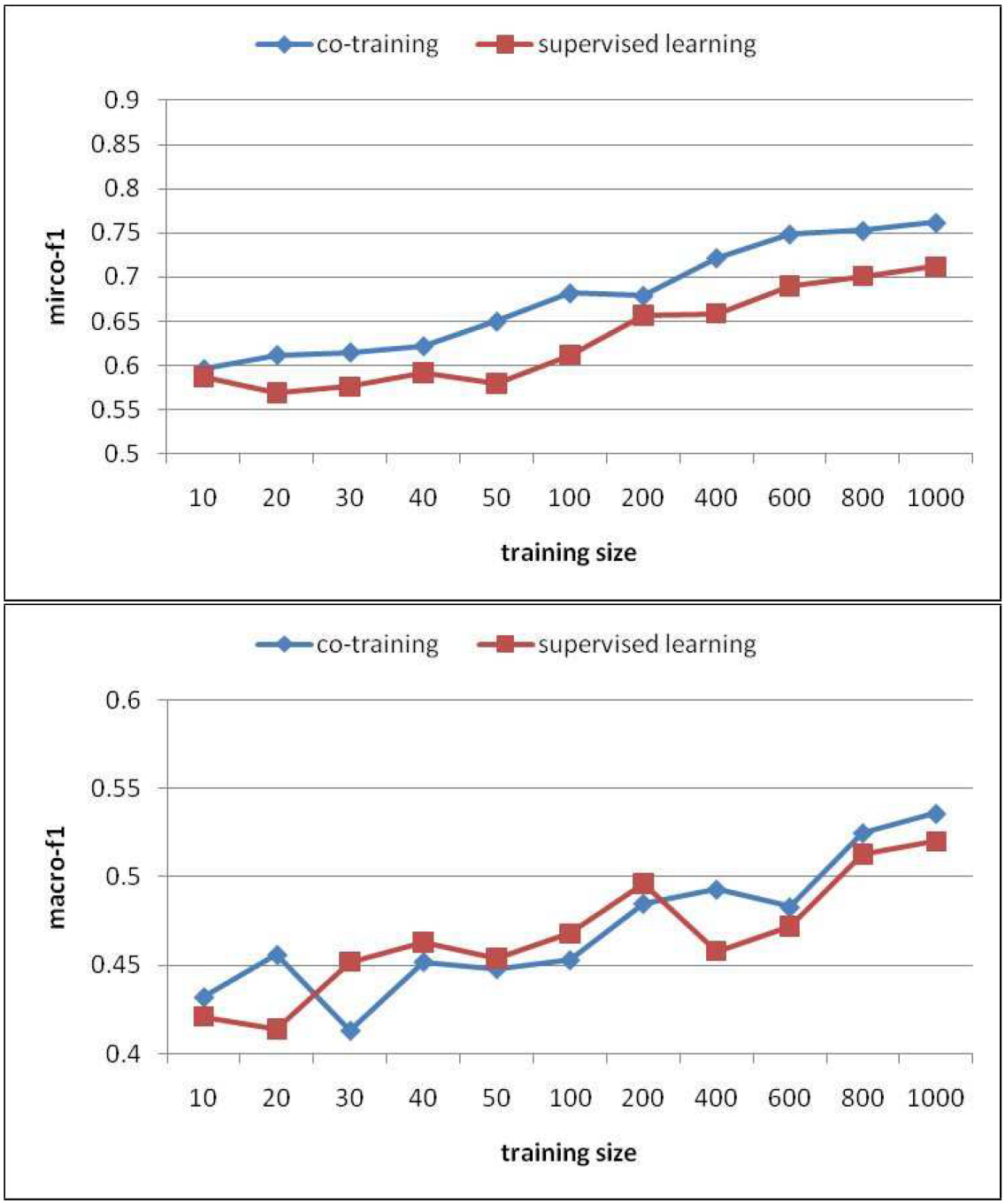

Figure 3.5: The performance of Co-Training vs supervised learning with varying number of labelled questions.

Table 3.1: The most discriminative textual features for each category of questions.

intent	textual feature	information gain
objective	`anyone`	0.096
	`what's`	0.087
	`who is`	0.054
	`why is`	0.054
	`what is`	0.044
subjective	`is your`	0.036
	`help`	0.026
	`can I`	0.014
	`favourite`	0.011
	`how do`	0.009
social	`anybody`	0.042
	`is there`	0.035
	`looking for`	0.028
	`do I`	0.028
	`I am`	0.011

Table 3.2: The dataset for experiments.

data	objective	subjective	social	*total*
training	503	442	70	1015
testing	259	228	37	524
all	762	670	107	1539

Table 3.3: The performance of supervised learning with different sets of features.

features	objective	subjective	social
text	0.693	0.689	0.152
metadata	0.609	0.642	0.378
text+metadata	0.731	0.693	0.412

Table 3.4: The performance of supervised learning vs semi-supervised learning (Co-Training).

approach	miF_1	maF_1
supervised (text+metadata)	0.712	0.510
Co-Training (text+metadata)	0.757	0.534

Chapter 4

Understanding User's Locality Intent

Many web sites involve businesses or information which provide services that are relevant to a specific location, such as the location of a restaurant or a theater in a certain town, or building postcode for a city. This chapter focuses on identifying user's locality intent by using a semi-supervised machine learning approach.

The rest of this chapter is organised as follows. In Section 4.1, we introduce the background of locality intent in CQA. In Section 4.2, we review the related work. In Section 4.3 , we define our taxonomy of user's locality intent in CQA. In Section 4.4, we introduce the PU approach to question classification with only positive and unlabelled examples. In Section 4.5, we describe the experimental setup and present our findings. In Section 4.6 , we conclude our work and contributions.

4.1 Overview of Locality Intent

Many information searchers submit their queries (whether or not they contain location key words) in such a way that makes it easy for a search engine to identify

relevant web sites. Unfortunately, unlike search engines, most current CQA services do not consider the user's locality intent, therefore user's information need is not satisfied geographically. For example, at the time of writing this chapter, the question "Whats the best restaurant to watch fireworks from in Hongkong?" attracts only one response from Yahoo!Answers, leaving a large margin space for the system to attract more Hongkong based users to answer it.

To shed light on the user's locality intent, we propose to classify questions into two categories according to the locality intent: local and global. By considering the question, for instance, "What's the best restaurant to watch fireworks from in Hongkong?" as a local one, a CQA system can route the question directly to some specific local answerers by identifying the corresponding spatial scope. On the other hand, by identifying "Where is a good place I can chat to people about money making ideas?" as a global question, we can highlight the question on the home page to attract more people answer it, regardless of their locality background. After performing the classification of local and global, we further pinpoint the spatial scope of the question by analyzing its thematic features so as to enable the search radius to vary depending on user's information need. For example, users querying for a coffee shop are probably looking for one within walking distance. If they are consulting the local tax rate, they will expect a distance of the nearest council. If they want to buy cheap ticket for travelling, however, distance may not be important as tickets can be bought over the Internet. CQA systems can then automatically pinpoint the specific locality scope by combining one's GPS coordinates and the spatial scope of the topic. This is a tempting scenario for a mobile environment: one can ask local question without explicitly mentioning their current location and intended search radius, producing a significantly enhanced user-experience in terms of simplicity and flexibility.

In this chapter, we build a predictive model through machine learning based

on both text and locality features to identify the locality intent. Our investigation reveals that the Probability Estimation model achieves a superior performance than S-EM and Biased-SVM. In addition to revealing the general locality intent, the spatial scope of the question is also further targeted and exploited. Our experiment shows that F_1 scores of 0.738 and 0.754 can be achieved on Yahoo!Answers and WikiAnswers datasets respectively (See Section 4.5.3 of this Chapter).

4.2 Previous Work on Locality Intent

The problem of understanding the user's locality intent was first proposed in the context of Web search engines. Luis et al. [25] classify the locality intent of Web search queries into two categories: global and local. However, this taxonomy is not that suitable for CQA, because web search engines aim to retrieve the most relevant web pages while CQA services strive to find the most appropriate people with the matching knowledge.

In the context of CQA, Zhou et al. [89] proposed a classification-based approach for question routing, which directs questions to answerers who are most likely to provide answers. They propose to use local and global features to enhance the classifier's performance. Li et al. [42] provide a question routing framework, which comprehensively considers user's expertise, availability and answerer rank by having these features integrated into a single language model. The motivation of this research is somewhat similar to ours, although none of them leverage user's geographical features.

With regard to the task of semi-supervised learning in CQA, Chapter 3 has already revealed that unlabelled questions are useful to improve the performance of question classification. In that chapter, we employ a Co-Training framework to identify subjective and social questions in CQA. However, as opposed to the Co-Training framework in which both positive and negative labelled examples are

compulsory for training, in this task we take advantage of the PU-Learning framework that only requires positive ones for training. To the best of our knowledge, this is the first CQA work that integrates PU-Learning framework in the designing of the system.

4.3 Research Problems Pertaining to Locality Intent

Taking into account the special locality characteristics of CQA, we propose the following taxonomy that classifies questions into two categories in terms of their underlying geographical locality: local and global. This allows us to transform the locality intent understanding problem into a local/global classification problem.

Local Questions The intent of such questions is to get information regarding a certain geographical locality, the best answers are likely to be produced by local answerers. For example, the question "Which country in Africa that was colonized by France did assimilation policy succeed?" asks for specific details of a particular location. Usually, local questions include one or more location names, as in the case of the question "What's the best restaurant to watch fireworks from in Hong Kong?", the asker tries to set up a connection with the Hong Kong community, from where the user can learn more details about a particular entity afterwards.

Global Questions The intent of such questions is to get information irrespective of the geographical locality, the best matches are usually general answerers. For example, the question "Why I cannot block someone on YouTube when there's a new channel design update?" asks information from a general trouble shooter regarding web site configuration, regardless of their local-

ity background. But more implicitly, the question "Is the Aurora Borealis phenomenon found anywhere else in the world?" may appear to be a local question (notice that Aurora Borealis always corresponds to the pole areas), until one comes to realize that there is no answerer available in such region.

Notice that our taxonomy is a two-level hierarchial structure, in which local category are further broken down into subcategories to pinpoint question's spatial scope. We inherit the administrative place types of Yahoo! Placemaker namely, *Country, State, County, Town, and Local Administrative Area* to further break down local questions into the second level of spacial scope. More-detailed information regarding different Places vs. Place Names can be found at the Yahoo! Placemaker Key Concepts page[1].

4.4 Approach to Dealing with Locality Intent

In the locality classification task, dozens of local questions can be automatically detected from location-based categories. For example, in the Dining Out category of Yahoo! Answers, questions have been broken down into city subcategories scattered around the world. On the other hand, however, it's impractical to label large amounts of global examples manually. Traditional supervised learning models are thus not helpful in the construction of an automated training model; they require training in both local and global examples. Therefore, we think that, the PU-Learning framework can fit in to this context quite well.

Basically, PU-Learning is a semi-supervised learning framework, which builds a classifier with only positive and unlabelled training examples, to predict both positive and negative examples in test dataset. A short introduction, which describes PU-Learning models, is given in the following sub-sections.

[1] http://developer.yahoo.com/geo/placemaker/guide/concepts.html#placesandplacenames

4.4.1 Spy-EM

Spy-EM model is first proposed in [47] and can be broken down into two steps. The first step is to identify reliable negative examples from the unlabelled set U, which works by sending some *spy* examples from the positive set P to U. The reliable negative examples are found through multiple iterations by running the first step a couple of times. The second step is to use EM algorithm to build the final classifier. However, the EM algorithm makes some mixture model assumptions [54] on the datasets, which can not be guaranteed to always hold.

4.4.2 Biased-SVM

The Biased-SVM [46] approach modifies the SVM formulation to make it fit in to the setting of PU-Learning, which can be described in the following SVM reformulations.

$$\begin{aligned} Minimize: \tfrac{<w \cdot w>}{2} + C_+ \textstyle\sum_{i=1}^{k-1} \xi_i + C_- \sum_{i=k}^{n} \xi_i \\ Subject\ to: y_i(< w \cdot x_i > + b) \geq 1 - \xi_i, i = 1, 2, ..., n \\ \xi_i \geq 0, i = 1, 2, ..., n \end{aligned} \tag{4.1}$$

In Equation (4.1), x_i is the input vector of the training example and y_i is its class label, $y_i \in \{1, -1\}$. The first $k-1$ examples are positive examples labelled 1, while the rest are unlabelled examples, which are treated as negative labelled -1. C_+ and C_- are parameters to weight positive errors and negative errors differently. We give a bigger value for C_+ and a smaller value for C_- because unlabelled examples, which assumed as negative, contains positive examples. The C_+ and C_- values are chosen by using a separate validation set to verify the performance of the resulting classifier.

4.4.3 Probability Estimation

Probability Estimation approach is famous for its prominent accuracy and distinctive computational simplicity. This approach was first proposed in [21] and utilizes some probabilistic formulas.

Denote x an example and y the binary label (local and global); let local questions be positive examples and global questions be negative ones. Let $s = 1$, if the example x is labelled, and $s = 0$ if otherwise. Thus, the condition that only positive examples are labelled can be described as:

$$Pr(s = 1|x, y = -1) = 0 \tag{4.2}$$

The formula (4.2) informs us that when $y = -1$, the probability of x being labelled is zero. So the objective now is to learn the classification function $f(x) = Pr(y = 1|x)$. To start with, the *selected completely at random* assumption has to be satisfied: the labelled positive examples are chosen completely at random from all the positive examples, and thus,

$$Pr(s = 1|x, y = 1) = Pr(s = 1|y = 1) \tag{4.3}$$

The training set consists of two parts: the labelled dataset P (when $s = 1$) and the unlabelled dataset U (when $s = 0$). Let $g(x) = Pr(s = 1|x)$ be the function that estimates the probability of an example being labelled, $f(x) = Pr(y = 1|x)$ be the function that estimates the probability of an example belonging to positive category. Then the following lemma shows how to derive $f(x)$ from $g(x)$

Lemma 1: suppose the "selected completely at random" assumption holds. Consequently,

$$f(x) = \frac{g(x)}{c} \tag{4.4}$$

The above equation suggests that we can attain a positive-negative classifier (this is exactly what we need) by having a positive-unlabelled classifier divided by

c: the probability that a random positive example being labelled. Notice that in Equation (4.4), $c = Pr(s = 1|y = 1)$ is a constant that represents the probability of positive examples being labelled. So the problem now lies in how to estimate the constant c by using a trained classifier g and a validation dataset. Three estimators are proposed in [21] namely, $e_1 = \frac{1}{n}\sum_{x\in P} g(x)$, $e_2 = \sum_{x\in P} g(x)/\sum_{x\in V} g(x)$, and $e_3 = max_{x\in V} g(x)$. In the above formulas, V is the validation datasets, P consists of all the labelled examples of V, n is the cardinality of P.

4.5 Experiments on Locality Intent

An unlabelled set and test set are randomly selected across all 26 Yahoo main categories of Yahoo! Answers. Note that as we leverage a PU-Learning framework in our task, the training set will only involve local questions. The training set is automatically extracted from the Dining Out, Travel, and Local Business categories with questions of a city name being assigned as the subcategory, whereas test set is manually labelled for both local and global examples.

The WikiAnswers dataset is collected from WikiAnswer[2] dating from 2012/01/01 to 2012/05/01 contains a total of 824320 questions (note that this is only a subset and cannot cover all the questions during that period of time). All the local questions are derived from the WikiAnswers Local category as we find this is the only category in WikiAnswers that is completely devoted to locality intent. We present the detailed statistics regarding the test and training sets, and validation set in Table 4.1. Acronym YA and WA represent Yahoo! Answers and WikiAnswers respectively.

With respect to the second-level classification, we use the same dataset by selecting all the questions containing at least one location reference (which is tagged by using Yahoo! Placemaker). There are 324537 and 12401 such questions available

[2] http://wiki.answers.com/

Table 4.1: Summary of CQA datasets

data	local	global	*total*
YA training	1000	0	1000
YA test	256	844	1100
YA validation	1000	0	1000
WA training	1000	0	1000
WA test	172	928	1100
WA validation	1000	0	1000
all	4428	1772	6200

in Yahoo! Answers and WikiAnswres datasets, respectively, which directly serves as the second-level datasets for classification. What's more, all the location references in the training set are hidden to emulate the scenario when mobile users forget to type in the specific localities.

4.5.1 Experimental Setup

Since the class sizes are imbalanced in this problem, we use the F_1 score instead of accuracy to measure the performance of question classification. The details regarding the F_1 score has been described in Chapter 3, Section 3.5.2.

4.5.2 Experimental Results

A number of machine learning algorithms implemented in Weka[3], including C4.5, Random Forest, Naive Bayes, k-Nearest-Neighbours, and Linear Support Vector

[3] http://www.cs.waikato.ac.nz/ml/weka/

Machine (SVM), have been tested for semi-supervised learning (PU-Learning). What we find is that SVM can constantly outperform other schemes so we use it as the basic learning scheme in the following subsections. In addition to text features, we exploit several locality features that can help in detecting the locality intent within the question. With the information annotated by Yahoo! Placemaker, the following sub-sections detail features that are considered in our framework.

4.5.2.1 Textual Features

The textual features of a question are extracted from the content of the question title after standard pre-processing steps (tokenization, lower-casing, and stemming), which is the same setting to that of Chapter 3, Section 3.4.1.

To have a rough idea about each category of questions, all unigram and bigram word features have been sorted in terms of information gain for question classification. We show the most discriminative ones in Table 4.2. It appears that questions with those location-related words are more likely to have a local intent, whereas questions with conversational phrases are more likely to have an unlabelled intent. This indicates that attributes regarding some location references in textual features may have relatively more power to separate local questions from the global ones.

4.5.2.2 Location Frequency

From Figure 4.1, one can see that the location frequency feature over Yahoo! Answers and WikiAnswers looks very similar with only around 2% differences. Questions with no location references are more likely to pertain to the unlabelled category, whereas questions with exactly one location are more likely to belong to the local category. This is quite intuitive: locality intent usually comes with location references. When it comes to the questions with more than one location references,

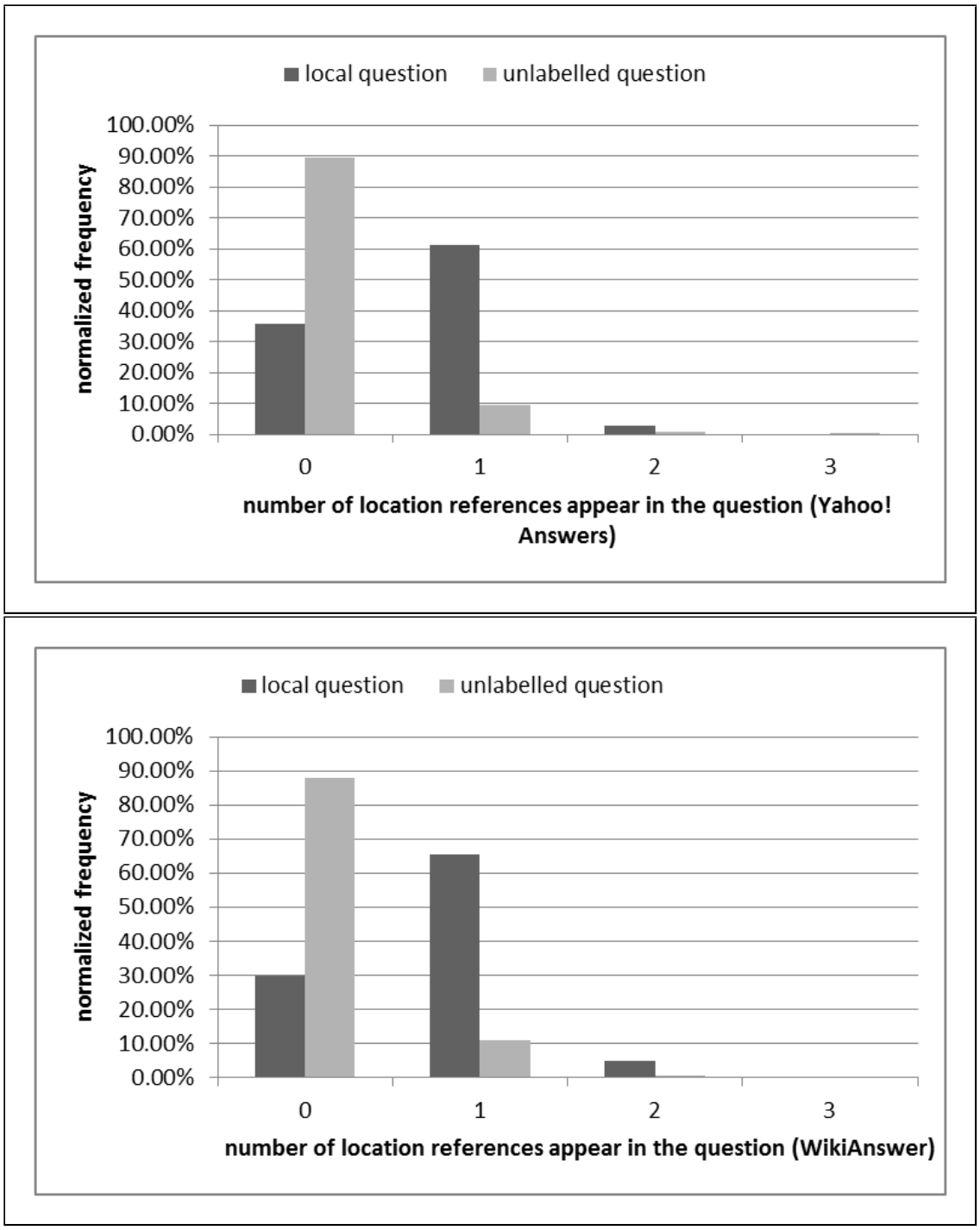

Figure 4.1: The location frequency feature over Yahoo!Answers (up) and WikiAnswer (bottom)

Table 4.2: The most discriminative textual features in Yahoo!Answers.

intent	textual feature	information gain
local	`best places`	0.0043
	`best way`	0.0027
	`anybody`	0.0025
	`between the`	0.0013
	`come from`	0.0009
unlabelled	`can you`	0.0049
	`cheapest`	0.0022
	`buy`	0.0022
	`deal with`	0.0014
	`changes`	0.0008

we did not observe any apparent patterns. We figure the reason is that questions containing more than one location generally suffered from the data sparsity problem and thus cannot serve as a good indicator.

4.5.2.3 Location Level

When two locations with different scope occur in one question, we use the lowest level of the scope as the question's representation — we believe users of small scopes have superior knowledge to cover those with the bigger scopes. The pattern we find in Figure 4.2 is that local questions tend to have advantage in Town scope while unlabelled questions take control of the State and Country scopes. It is probable that local questions are more likely to occur at a local level namely — town, while

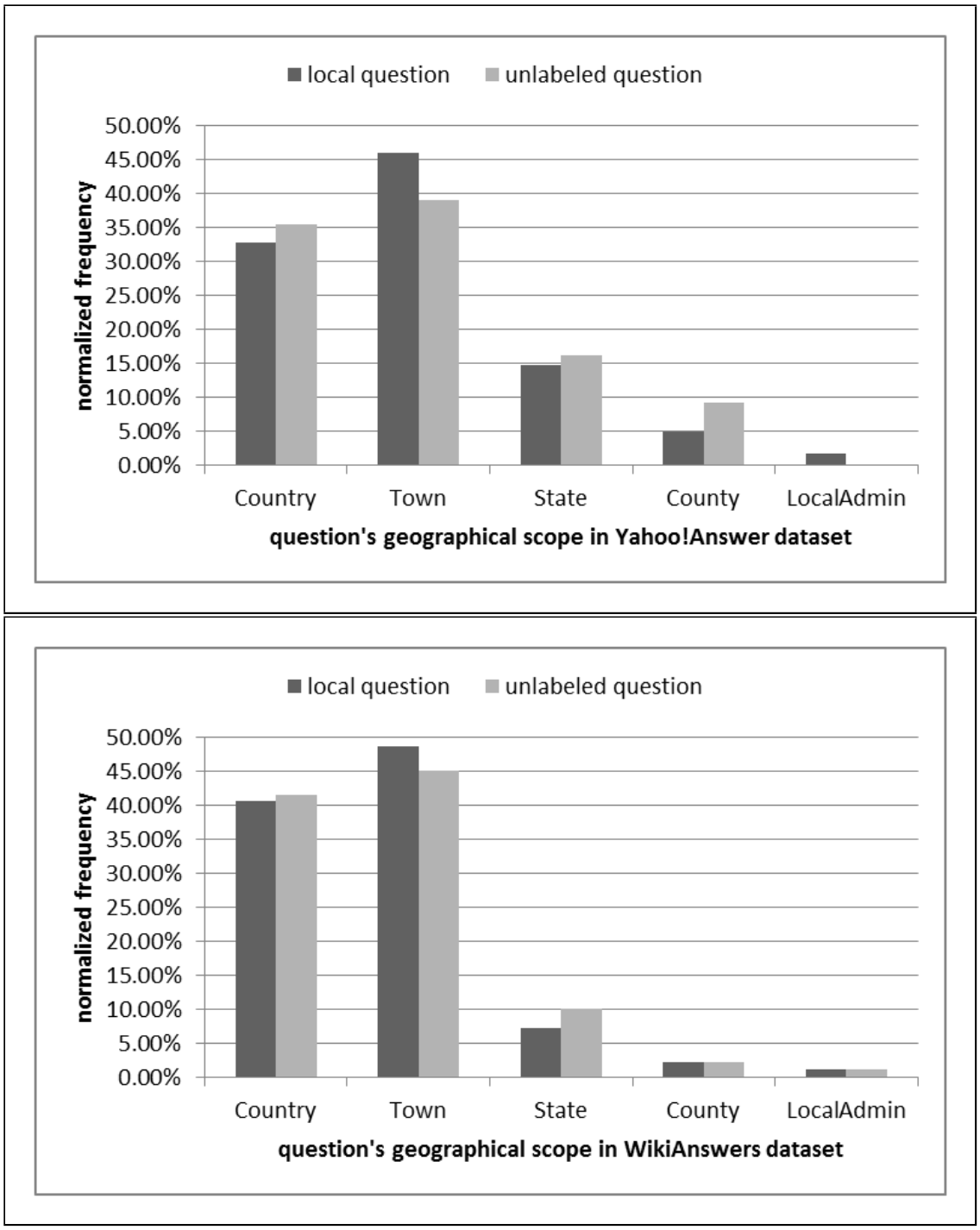

Figure 4.2: The location scope feature over Yahoo!Answers (left) and WikiAnswers (right)

unlabelled questions are slightly more likely to appear at a general level, such as state. But we did not manage to reach a consistent result over the two datasets for County and Local Administrative Area scopes due to the high variance among small training examples.

4.5.2.4 Semi-Supervised Learning

We exploit several locality features that can help detect the locality intent within the question, namely location frequency and location level, in addition to the textual features. We note that the original question datasets are not geographically annotated and contain no locality information. Therefore, in order to extract location references and assign geographical scope to each question, Yahoo! PlaceMaker was employed to augment original datasets with the location-specific explanation. There are two versions of scopes available in Placemaker, namely the geographical scope and the administrative scope. Geographic Scope is the place that best describes the document and may be of any place type. Administrative Scope is the place that best describes the document and has an administrative place type (which refers to Country, State, County, Local Administrative Area and Town). We use the geographical scope in this chapter because we find this version provides more detail than administrative scope[4].

Figure 4.3 shows the learning curve of the PU-Learning schemes given a varying number of positive labelled examples (the unlabelled examples are fixed at 5000) in Yahoo! Answers. Figure 4.4 is the same learning curve for Wiki Answers as that shown in Figure 4.3. We employ the S-EM scheme to serve as baseline and the Biased-SVM as the state-of-the-art. As far as we can tell from the miF_1 figures, the two datasets share a similar result, in which Probability Estimation and Biased-SVM perform significantly better than S-EM given sufficient amounts of labelled

[4] http://developer.yahoo.com/geo/placemaker/guide/concepts.html

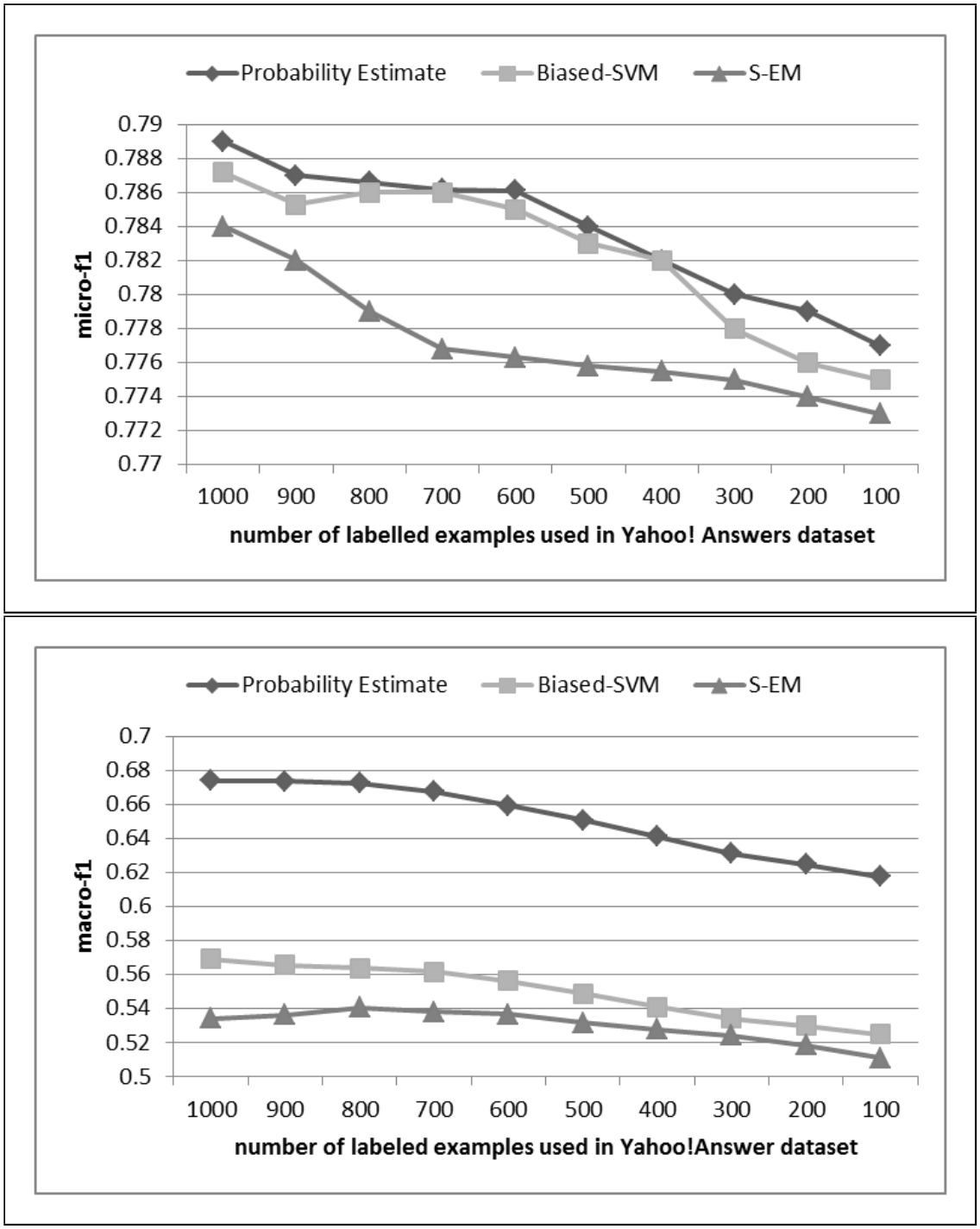

Figure 4.3: The micro F_1 (top) and macro F_1 (bottom) of PU-Learning with decreasing number of training examples used in Yahoo! Answers

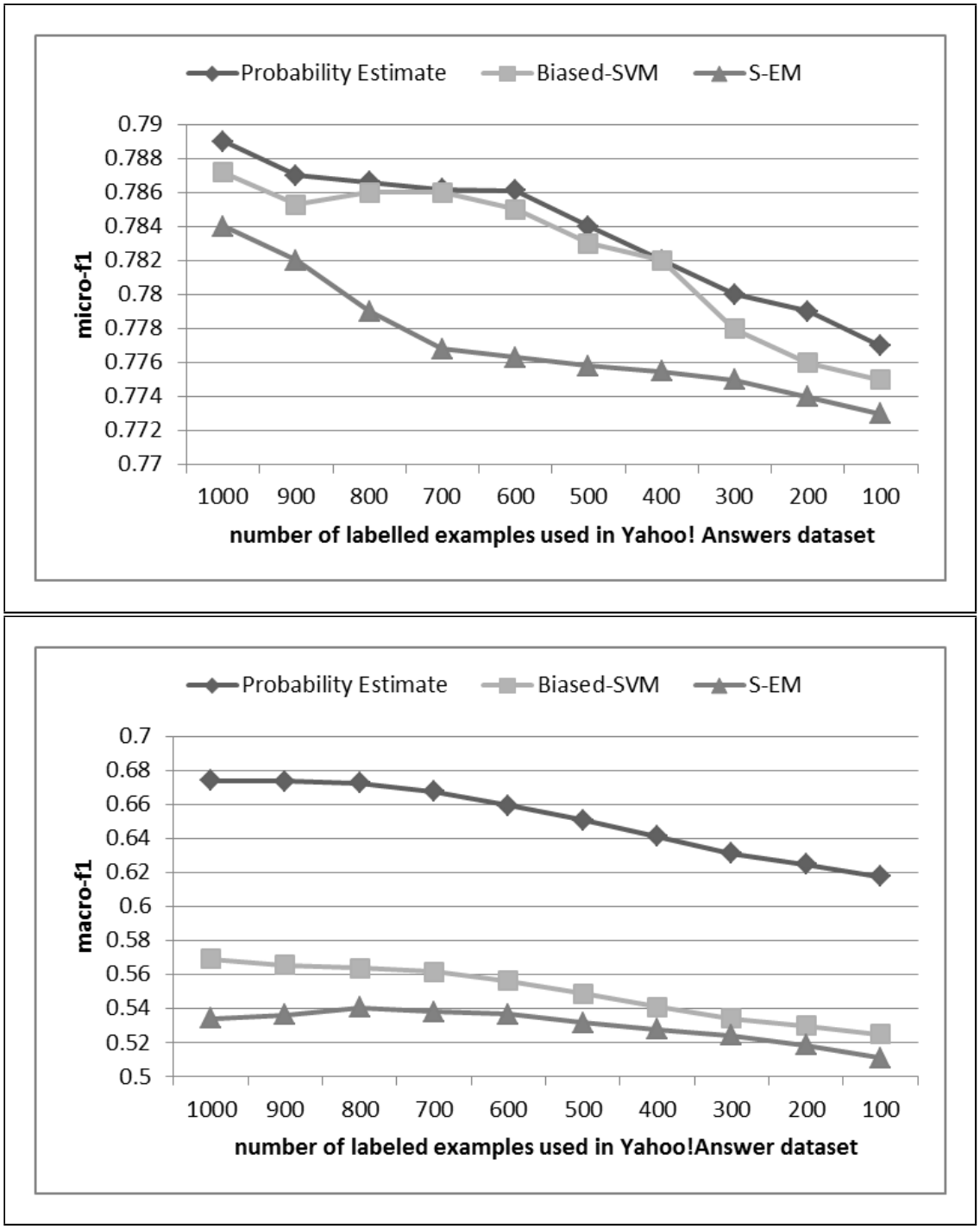

Figure 4.4: The micro F_1 (top) and macro F_1 (bottom) of PU-Learning with decreasing number of training examples used in Wiki! Answers

examples. However, the gap starts to decrease when we shrink the labelled size. All three approaches give a comparable performance when providing only 500 labelled examples or less.

As for maF_1 figure over the YA dataset, Probability Estimation consistently outperforms the other two schemes, an approximately 23% error reduction on the basis of Biased SVM, irrespective of the labelled data size; At the same time, Biased-SVM is slightly better than S-EM approach with an average 2% improvement. The result generated on the WA dataset for maF_1 is quite similar. We propose that the probability approach can overwhelm the other two due to the uneven distribution of the test set: 20% positive examples vs. 80% negative ones. In Probability Estimation model, having the non-traditional classifier divided by a constant, c, that enables the classifier to be more tolerant towards the positive classifying errors by sacrificing some negative examples. We believe that is why Probability Estimation, in some cases, is even slightly worse than Biased SVM under miF_1, producing a superior result for maF_1 by picking up the minority class in general.

4.5.3 Predicting Spatial Scope

We use the SVM implemented by Platt et al. [63] with a probabilistic output and adopt a linear kernel in this task. The setting of the classifier is similar to that of Chapter 3, Section 3.5.3.

Table 4.3 gives the result of the maF_1 and miF_1 comparison over each scope level. Under the evaluation of maF_1, the prediction on country, town and state scopes have a superior performance than the rest, this suggest that these three scopes are relatively easier to identify by inferring the question's topic (for both Yahoo! Answers and WikiAnswers). However, the system only displays mediocre performance regarding county and local administrative area scopes, which leads

Table 4.3: The F_1 of each scope category

data	country	town	state	county	admin	average
maF_1(YA)	0.713	0.684	0.670	0.497	0.363	0.585
maF_1(WA)	0.729	0.703	0.625	0.458	0.260	0.555
miF_1(YA)	0.818	0.693	0.474	0.215	0.131	0.738
miF_1(WA)	0.833	0.703	0.324	0.183	0.177	0.754

to our speculation that the questions in a higher scope level may have more discriminative power than questions in lower scope level. This is quite explainable, the questions with a larger scope tend to have generalization behaviour whereas questions with smaller scope are liable to have uniqueness behaviour. Under the evaluation of miF_1, the performance over Yahoo! Answers and WikiAnswers are 0.738 and 0.754 respectively, which suggests that majority of the local questions' scope can be accurately predicted even if user does not mention the place names.

4.6 Summary

The main contribution of this chapter is twofold. First, we identify several locality features which can be used together with standard textual features by machine learning algorithms to classify questions according to their geographical locality. Second, we prove that Probability Estimation approach can consistently outperform the S-EM and Biased-SVM on the evaluation of maF_1 and miF_1. Third, we prove that the spatial scope of a local question can be inferred accurately even if it does not mention any place name.

Chapter 5

Understanding User's Navigational Intent

Many questions in CQA can be resolved by external web pages which are already available on Internet, and thus it is useful to identify these questions to facilitate the performance of search engines and CQA services. This chapter focuses on understanding the user's navigational intent by employing a supervised machine learning approach, and demonstrating how to exploit it to evaluate the performance of search engines. The rest of this chapter is organised as follows. In Section 5.1, we introduce the background of navigational intent in CQA. In Section 5.2, we review the related work regarding navigational intent in CQA.. In Section 5.3, we give detailed definitions of users' navigational intent. In Section 5.5, we evaluate the performance of current search engines for handling verbose queries. In Section 5.4, we investigate the usefulness of text and metadata features for identifying the user intent of questions by using a supervised machine learning approach. In Section 5.6, we present our conclusion.

5.1 Overview of Navigational Intent

A vast majority of search engine queries are very short. For example, the average query length of an MSN search was 2.4 words [22]. However, there is also a non-negligible proportion of long queries, about 10% of queries are 5 words or longer [22]. Current search engines present convincing performance over short keywords queries but usually fail to handle verbose or colloquial queries competently [30].

However, verbose queries can be found in CQA services. It is difficult to encourage users to answer difficult questions in CQA, especially for those information-driven ones, since answering informational questions requires certain in-depth knowledge that only a small proportion of the population have the capacity of resolving it. Enabling search engines to answer verbose queries efficiently and effectively may remove the needs of submitting navigational questions to CQA services.

In this chapter, we endeavor to address the following two questions:

- *What is the performance of current search engines in handling navigational questions?*
- *Can we identify navigational questions from CQA services automatically?*

We define questions resolved (or largely explained) by their linked web pages (i.e., in the corresponding answers) as navigational questions, which are simulated as verbose queries for the search engine evaluation. The rationale is that queries from CQA services are less artificial when compared with TREC QA queries and less constrained when compared to search queries, where users are prone to generate queries in a simple keyword style. However, due to the inhomogeneous nature of the CQA services, questions cannot be treated as navigational questions directly. For example, as revealed by Chen [19], that 43% of questions in CQA are of subjective intent and 10.2% are of social intent. To solve this problem, Huston et al. [30] use a method in which they consider queries from certain categories as verbose queries,

which are then submitted to search engines. This method is effective in filtering short web-style queries, however, it may fail to remove the question with subjective (sentiment-based) opinions or social interactions intent. In this chapter, we use the dichotomy of navigational versus non-navigational, in which navigational questions can be resolved by (or at least largely explained by) web information while non-navigational questions usually require participants in the community to answer them manually.

Automatically identifying navigational intent of a new question is not an easy task since it is hard to recognize navigational intent by textual features. For example, the question "Can anybody recommend decent free music creation software?" with a survey style seems to have a transactional intent , but it is actually a navigational question with the best answer like "Hyrogen is ok, http://www.hydrogen-music.org/ ...." This implies that navigational intent is not always easy to be inferred solely based on textual features. Rather, metadata features, such as the asker's asking experience or the category from which the question corresponds to, is crucial for the intent deduction. Thus we build a predictive model through machine learning based on both text and metadata features.

5.2 Previous Work on Navigational Intent

Current search engines have been evaluated in various ways. Liu[1] assesses the effectiveness of Google, Bing, and Blekko by surveying 35 undergraduate students in Computer Science, from which he concludes that Google and Bing share a comparable performance in 2011. Liu et al. [51] provide a comprehensive study on predicting user satisfaction in CQAs and discuss how to evaluate it through machine learning. The work most similar to ours is [30] in which Huston et al. use Yahoo! Answers questions to evaluate search engine performance with the Yahoo! API and the Bing

[1] http://www.cs.uic.edu/~liub/searchEval/Search-Engine-Evaluation-2011.pdf

API respectively, and they find Bing is slightly better than Yahoo in 2011. Our approach is somewhat similar to [51], but instead of evaluating the relevance of documents with human judgment, we propose to automate the evaluation process by matching between the associated URLs in the answers and the search engine results. It may not be as accurate as human evaluation, since not all the associated URLs are good answers, and a large number of relevant web pages may be omitted. However, our approach contains a substantial number of questions we can process (especially when we can obtain an unlimited number of questions from CQA sites), which cancels out the side-effects of the incomplete judgment.

With regard to the task of navigational intent identification, Broder's seminal paper [12] divides the intent of web search queries into three categories: informational, navigational, and transactional. Lee et al. [41], later on, proposed a framework to automate the process of navigational intent identification in web search, in which *user-click behavior* and *anchor-link distribution* features are found to be useful for detecting navigational intent. Sadikov et al. [69] model the user's navigational intent by clustering document clicks and session co-occurrence information. However, these models cannot be directly applied to CQA due to the different expectations within people's mind-sets: in CQA users normally ask natural language questions which are addressed to human beings, whereas in Web search users submit keyword queries which are addressed to automated search engines. To the best of our knowledge, this is the first attempt to understand user's navigational intent in the CQA setting.

5.3 Research Problems Pertaining to Navigational Intent

Google is arguably the most powerful search engine in the world, and Bing has been rising up enormously recently, both of which are proved to be viable searching paradigms. Which one is a better choice is still one of the most controversial topics in the IR community. In light of this, we experiment with the search engines for dealing with navigational questions derived from Yahoo! Answers.

A vast amount of navigational questions is available on CQA services. Indeed, in 2005, 11.5% of questions in Yahoo! Answers have at least one URLs in one of the answers and 5.5% of questions include at least one URL in the corresponding best answer. Users cannot access the linked page themselves either because they don't have the necessary search optimization skill or they prefer communicating with people rather than the text produced by search engines.

The following examples illustrate navigational questions that askers currently post on Yahoo! Answers:

- **Navigational:** What is the best free online photography portfolio website? *I want to get into photography. is there a free online portfolio that prevents people from being able to right click and save the pictures?*
- **Non-navigational:** How much should you tip a pizza delivery man?

5.4 Experiment on Navigational Intent

To address the task of navigational question prediction in CQA, a variety of personal information and social relationship features are collected and exploited to model the users' social behavior behind their search intent.

5.4.1 Setup

The classification experiment is based on Yahoo! Answers dataset which is derived from Yahoo! Answers Comprehensive Questions and Answers (v1.0), a dataset kindly provided by Yahoo Research Group[2]. The details regarding this dataset can be found in Chapter 4, Section 4.5.

5.4.2 Classification Performance Measure

Since the class sizes are imbalanced in this problem, we use the F_1 score [53] instead of accuracy to measure the performance of question classification. The details regarding the F_1 score has been described in Chapter 3, Section 3.5.2. Note that there are two versions of F_1 score, namely maF_1 and miF_1, the results reported in the next section are all predicated on (maF_1).

5.4.3 Textual Features

The textual features of a question are extracted from the bag-of-words content of the question title after standard pre-processing steps (tokenization, lower-casing, and stemming) [53]. Therefore, each question is represented as a vector of terms weighted by TF×IDF [53]. We didn't remove stop words since we found that stop words slightly improved the classification performance.

To have a rough idea about each category of questions, we sort unigram and bigram features (words that occur in the question) in terms of information gain for question classification, and show the most discriminative ones in Table 5.1. It is clear that questions with those web sites words (site, download, and email) are more likely to have navigational intent, whereas questions with conversational phrases are more likely to have non-navigational intent. But the information gain

[2] http://webscope.sandbox.yahoo.com/

Table 5.1: The most discriminative text features for each category of questions.

navigational features	information gain	non-navigational features	information gain
where can	0.0098	your	0.0037
find	0.0092	is your	0.0029
download	0.0087	anyone know	0.0023
web site	0.0087	can	0.0023
where I	0.0079	if	0.0019
best	0.0068	photograph	0.0018
I find	0.0046	in what	0.0018
good website	0.0038	answer for	0.0016
I can	0.0033	for my	0.0015
email	0.0029	the history	0.0015

values for textual feature is relatively low, which suggests that textual features have a weak discriminative power to separate navigational from non-navigational.

5.4.4 Question Topic

Figure 5.2 (top) depicts the distribution of user intent over the top-10 navigational question categories. One can see that navigational questions have a small presence in most categories except for "Games Recreation", "Computer Internet", and "Business and Finance", where their presence is higher. A possible reason for the first two categories is that they are more concerned with Internet-based information than the rest of the categories, and therefore answerers are incline to steer users to the pertinent web resources. But what surprises us is that "Business and

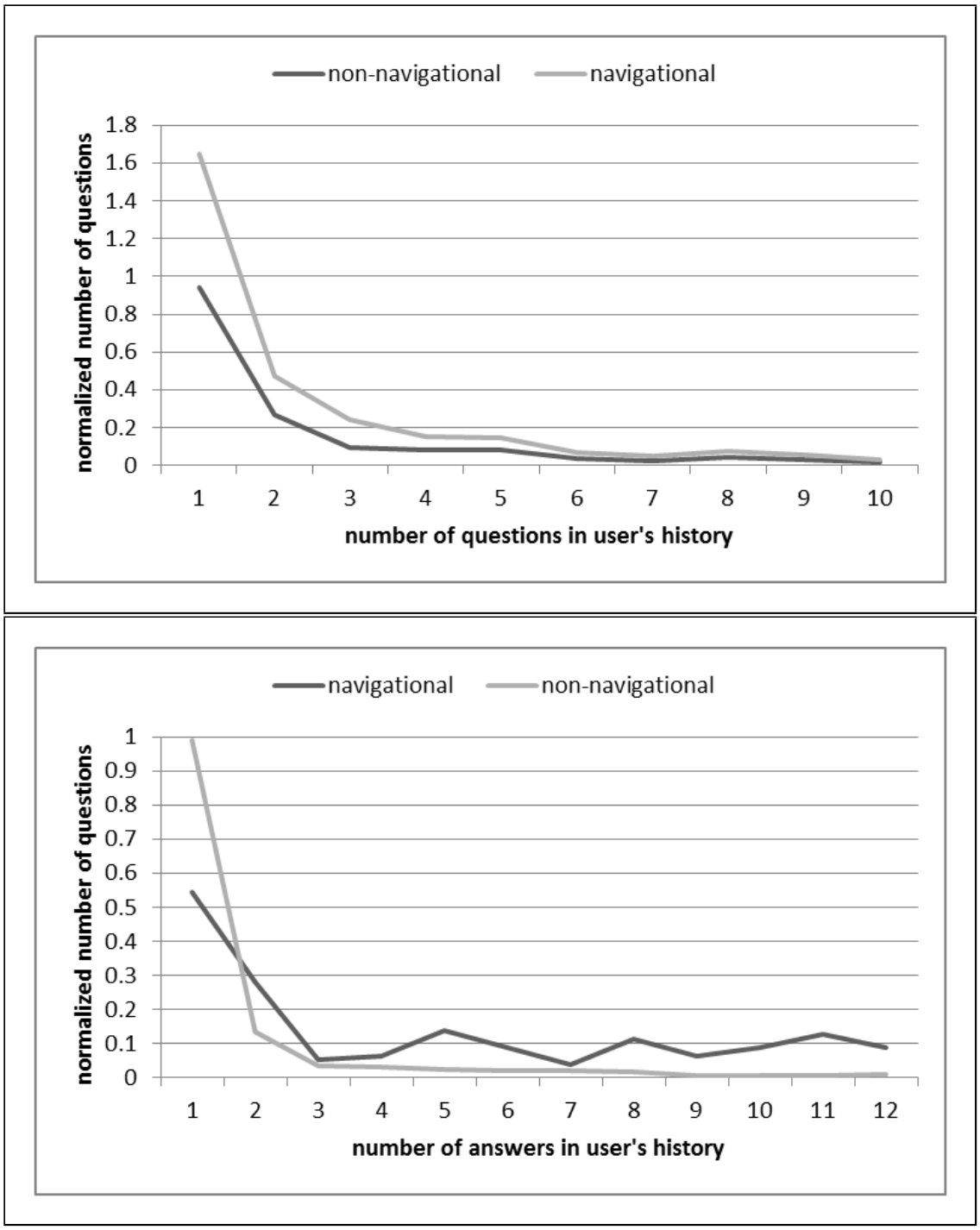

Figure 5.1: The asking experience feature (top) and the answering experience feature (bottom)

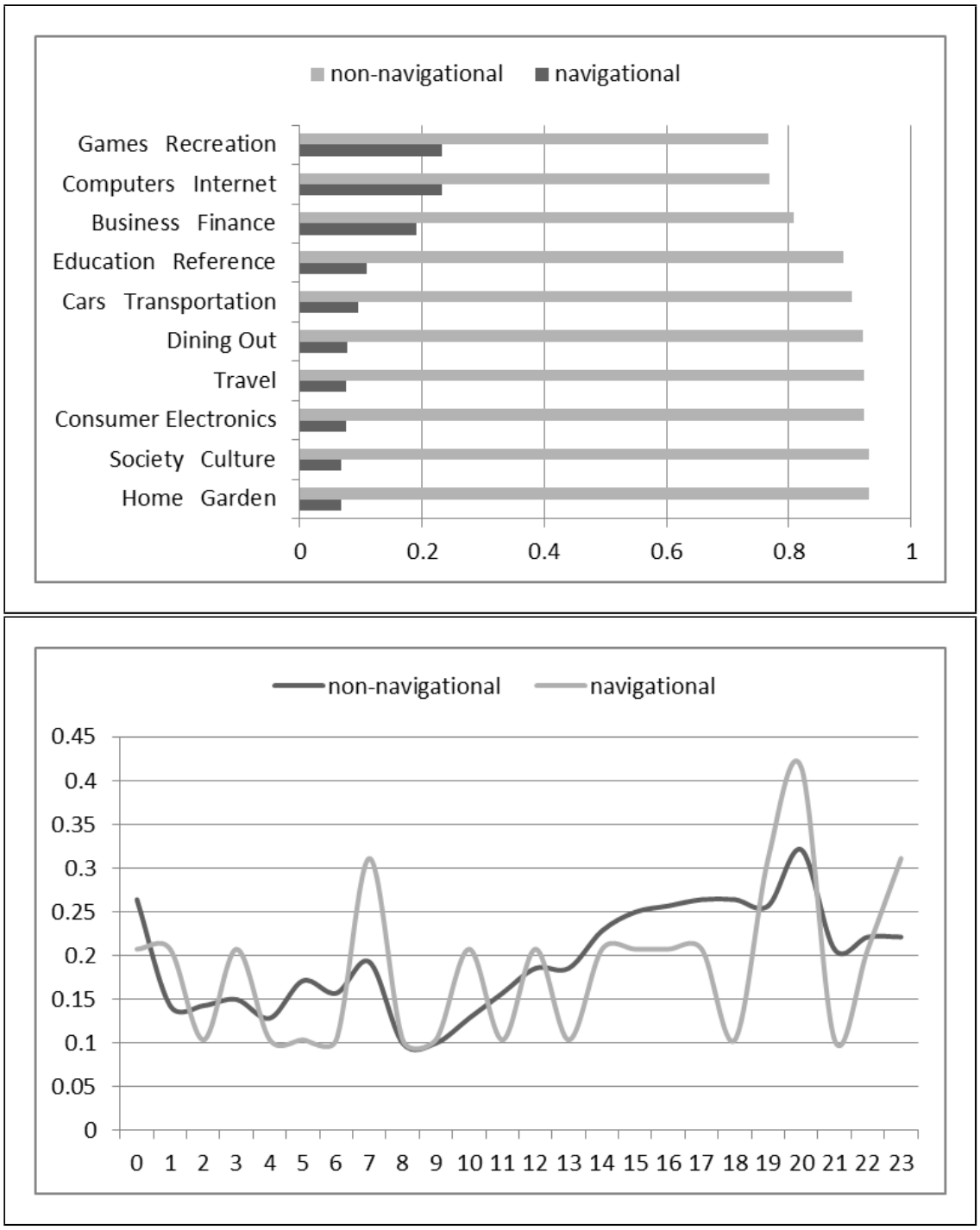

Figure 5.2: The question topic feature (top) and the question time feature (bottom)

Finance" also contains a high percentage of navigational question. After checking some samples, we came to realize that it is due to the fact that there are many transactional questions within this category, as in the case of the question: "I need to locate a man enlisted in the Army; I have his SSN, but no station location. How can I find him?"

5.4.5 Question Asker Experience

Figure 5.1 (top) shows the distribution of user intent over the question asker's asking experience (i.e., the number of questions the user has asked before.) It seems that experienced users are more inclined to ask non-navigational questions, perhaps navigational questions are usually more boring than non-navigational one so that users tend to get negative feedback when asking navigational questions.

Figure 5.1 (bottom) shows the distribution of user intent over the question asker's answering experience, which refers to the number of questions the user has answered before. This is consistent with the results of asking experience, users who spent more time on Yahoo! Answers are more likely to ask non-navigational questions while navigational questions are more likely to be asked by novices.

5.4.6 Question Time

Figure 5.2 (bottom) shows the distribution of user intent over the time (hour-of-the-day) when the question was asked on 1st May 2006. Navigational questions show interesting patterns: the peak time for navigational questions is at 7:00 (starting the day-time work), 20:00 (after dinner), and 23:00 (about to sleep).

Table 5.2: The most discriminative metadata features.

metadata feature	information gain
`question topic`	0.3129
`question asker's experience`	0.0976
`number of answers`	0.0437
`question time`	0.0320
`weekend/weekdays`	0.0170

5.4.7 Metadata Features Results

To gain insight with regard to which metadata features are more informative for the identification of navigation-intent, we calculate and sort the information gain for top 5 metadata features used in our experiment, which is reported in Table 5.2. Consistent with our intuition, the question topic feature is arguably the most informative feature since it provides deeper and details-specific information about the question subject. Question asker's experience and number of answers features are good indicators of the question quality, and thus have a distinctive informativeness. Question time and weekend/weekdays features contribute evidence to modelling user's search behavior, but appears to be less important compared with the prior features.

5.4.8 Classification Results

We use SVM as implemented by Platt et al. [64] with a probabilistic output and adopt a linear kernel in this task. The setting of the classifier is the same to that of Chapter 3, Section 3.5.3. The parameter for the class weights is set as $navigational : non - navigational = 0.9 : 0.1$ since the classification task is an

Table 5.3: The performance of supervised learning with different sets of features.

features	non-navigational	navigational
text	0.873	0.363
metadata	0.934	0.883
text+metadata	0.936	0.893

imbalanced problem in its nature.

Table 5.3 depicts the performance (maF_1) of [binary] question classification through supervised learning (linear SVM) with different sets of features, by using 10-cross validation. It was quite surprising to us that the metadata features are even more important than textual features by giving insight of the user's asking behaviors. However, the mixture classifier with both text features and metadata features works better than the textual features classifier or metadata features classifier on their own, which only look at one perspective of the user intent.

5.5 Approach to Dealing with Navigational Intent in Search Engines

In this section, we conduct a experiment which tests both search engine's ability to answer the navigational questions of Yahoo! Answers.

5.5.1 Setup

The search engine evaluation experiment is derived from a dataset crawled by ourselves, which is collected from Yahoo! Answers[3] dating from 2013/03/15 to 2013/04/01, contains a total of 54483 questions (note that after data cleansing,

[3] http://answers.yahoo.com/

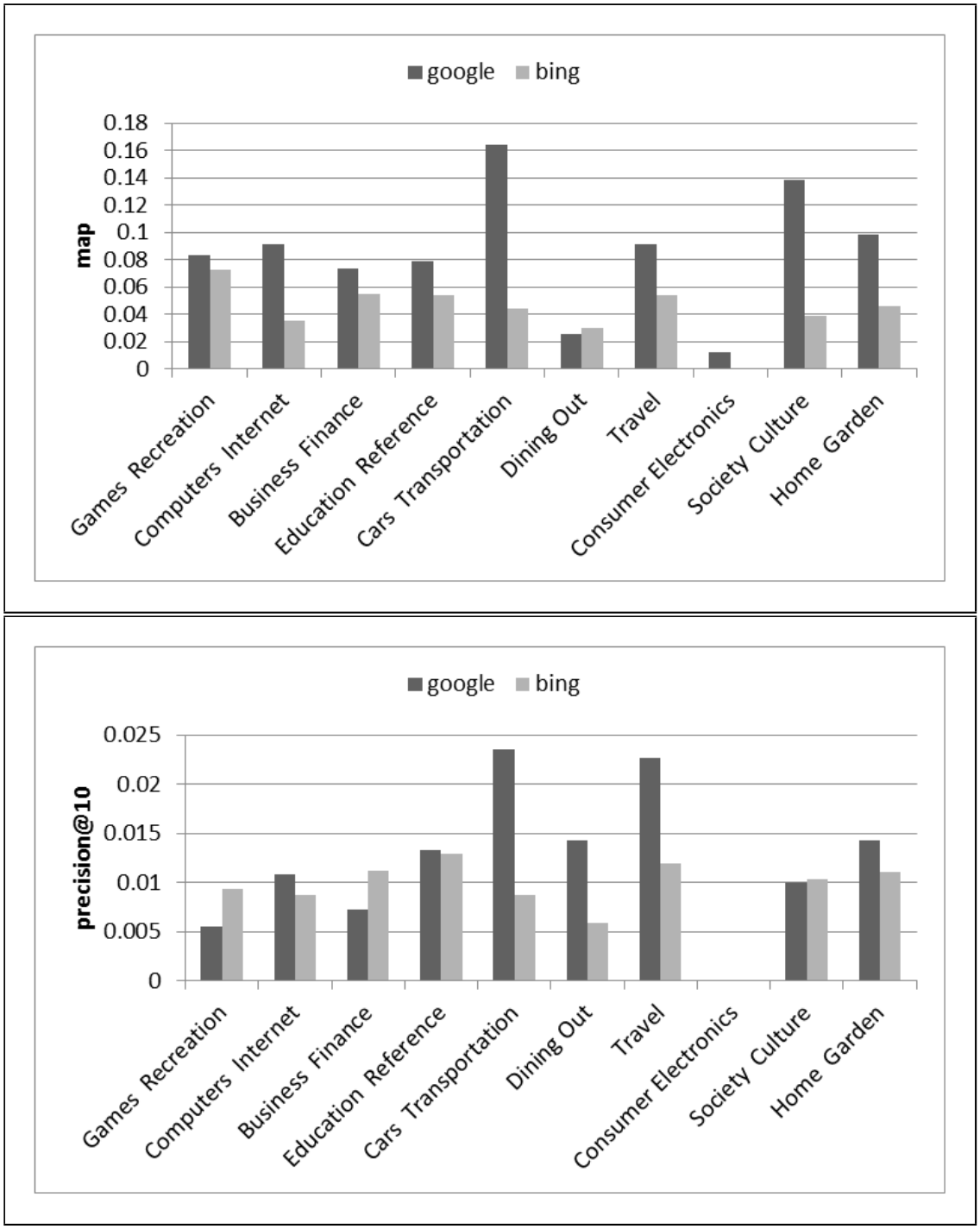

Figure 5.3: The performance comparison between Bing and Google for dealing with verbose questions over top 10 Yahoo! Answers navigational categories.

this is only a subset and cannot cover all the questions during that period of time). We adopt this dataset for the search engine evaluation task because they are collected fairly recently and should have been well indexed by both Google and Bing. There are 5747 navigational questions in this dataset, from which 3752 ones are from top 10 Yahoo! Answers categories which are then simulated as test data to evaluate the search engines.

Google API[4] and Bing API[5] were employed for evaluation because we observed that the "black box" approach has been extensively used in many recent research papers [26, 30] and is becoming more and more important for commercial purposes.

5.5.2 Stopword Removal

There are many stopword lists available in the IR community, but we chose to create our stopword list since the language that used in the test questions is more noisy than a regular English text. We adopt an IDF-weighting scheme to the Yahoo! Answers repository to assist us in stopword removal. Specifically, we construct a stopword list by taking the top 100 words from the inverse document frequency ranking. This process identified words such as "help", "anyone," and"what", which may not appear in the standard stopword list, but are usually not useful as search terms.

5.5.3 Noun Phrase Detection

Learning from the previous research that noun phrases from the query can help identify the key concepts within the query, we used the Standford Parser toolkit [39, 67] to automatically extract those potential noun phrases. Considering that we are

[4] https://developers.google.com/web-search/
[5] http://datamarket.azure.com/dataset/bing/search

using a search engine as a black box, the usage of the noun phrase technique is a restrictive form of query: it is impossible to assign weights to terms in terms of condence or priority. There are many ways to enable a search engine to communicate with the extracted noun phrases, we report on two such methods:

- The first method put each of the extracted noun phrases in the query in quotation marks, removing no words in the query.
- The second method is to only keep the extracted noun phrases and quotation marks are not used.

For example, 2 noun phrases: "the website" and "American eagle" are detected in the query:

"what is the website for American eagle?"

Using the first method, we would generate the query:

what is "the website" for "American eagle"?

Using the second method, we would generate the query:

"the website" "American eagle"

5.5.4 Search Results

The retrieval performances, measured by *Precision at 10* (P@10) [53] and *Mean Average Precision* (MAP) [53], are reported in Table 5.4 and Figure 5.3. For relevance judgement, only the URLs appeared in the answers are regarded as relevant web pages. Note that we employ MAP instead of MRR (Mean Reciprocal Rank) because there are often several URLs appearing in the answers such that the number of relevant web pages is usually uncertain. Even though the relevance judgments for the verbose queries are incomplete and the absolute retrieval performance is relatively low (which is expected because of the sparseness of the relevance judgment), our approach is probably more reasonable than traditional ones since it has

Table 5.4: Summary of the search engines evaluation for dealing with verbose queries (statistical significance using Paired t-tests were performed between each result shown and the Original: ** indicates p-value < 0.01 while * indicates p-value < 0.05).

	Google API		Bing API	
	map	precision@10	map	precision@10
original	0.0687	0.0112	0.0452	0.0101
stopwords removal	0.071*	0.0124*	0.0467*	**0.0115****
quoted noun phrases	0.0457	0.0089	0.0372	0.0075
only noun phrases	0.0715*	**0.012****	0.0475*	0.011*
quoted noun phrases + stopwords	0.0472	0.0109	0.0412	0.0083
only noun phrases + stopwords	**0.0732****	**0.013****	**0.476****	**0.0114****

been demonstrated by Carterette [16, 17] that evaluation over more queries with fewer or noisier judgments is preferable to evaluation over fewer queries with more judgments. The large number of the test data compensates for the incompleteness of the judgments. Another concern is that the search results may be time-sensitive since most search engines are regularly updated on a hourly basis. In order to reduce this risk, we submitted all queries of the above approaches to search engines within a short time session, spanning from 26/03/2013 to 30/03/2013. One should also note that search engines often return the Yahoo! Answers original web pages, which were removed from the results to allow an impartial judgment.

Table 5.4 reports the retrieval results for all of the query processing techniques when applied to Yahoo! Answer test data using Google and Bing. The results from the two search engines are very similar in terms of precision@10; when

it comes to *MAP (Mean Average Precision)*, however, Google overwhelms Bing with almost 50% improvement. This suggest that Google and Bing have a comparable ability to capture the desired documents, but Google is superior to Bing when ranking user's desired documents. Also the use of quotations of the noun phrases (method one) for the query reformulation is clearly not effective. But both noun phrase (method two) and stopword removal produce significant improvements. The most effective technique, however, is the combination of the above two.

Some users may be curious about which search engine is more capable of searching which topics (especially for those working in the advertisement industry where people need to strategise their investment smartly). For that reason, we also present a separate performance comparison under each top 10 Yahoo! Answers navigational categories. Although most of the categories share a comparable performance in Figure 5.3, it is clear that Google excels in *Car Transportation, Travel*, and *Home Gardens* categories, whereas Bing can hardly beat Google for any categories (some categories show inconsistent results over *precision@10* and *MAP*, such as *Business Finance*).

5.6 Summary

The contribution of this chapter is two fold. First, to our knowledge, this is the first work which attempts to understand user's navigational intent in CQA. Second, we propose a novel evaluation method which automates the verbose query evaluation process by matching the associated URLs in the answers (of the navigational question) and the search engine results. The current best search engines, namely Google and Bing, are evaluated with navigational questions (acting as verbose queries), from which we find that Google still outperforms Bing. In addition, we find that the best way to achieve query refinement for the current search engines is to combine both noun phrases (method two) and stopword removal techniques.

Chapter 6

Understanding User's Procedural Intent

Users often ask questions which require answers regarding certain procedures, for example, when they need to know how to accomplish a certain task. This chapter will focus on how to understand such procedural intent in CQA. The rest of this chapter is organised as follows.

In Section 6.1, we introduce the overview of procedural intent in CQA. In Section 6.2, we review the related work. In Section 6.3, we define *how-to-questions* and identify several patterns to extract them from Yahoo! Answers. In Section 6.4 , we introduce the two-stage framework for answering *how-to-questions*, and we investigate the usefulness of various features. In Section 6.5, we describe the experimental setup and present the experiment results. In Section 6.6, we present our conclusions.

6.1 Overview of Procedural Intent

As mentioned in Chapter 1, Section 1.2, Automatic QA is deemed to be the "Holy Grail" of QA research, since it can remove the need of submitting the question for human answering by steering askers to access the pertinent text from an immense body of knowledge. While significant progress has been made for resolving factoid questions (which has been discussed in Chapter 1, Section 1.2), answering more challenging non-factoid questions — such as *how-to-questions* — is still in its infancy since their answers cannot be found by simply employing the results of search engines. Given the complexity of resolving non-factoid questions, this chapter endeavours to answer one of the principle types, i.e., questions with the procedural intent.

Answering *how-to-questions* is a difficult task since they often bear task-specific information needs which require the answerer to have a good and detailed understanding of the question subjects. To study the potential effectiveness of using external resources, for example, eHow, to answer a new how-to-question, we carry out our analysis on three active categories of Yahoo! Answers, namely *Pets, Health*, and *Travel* (We use these three categories because they are also available in eHow). More specifically, we extract a subset of *how-to-questions* asked in these three categories, and validate whether they have a good match from eHow questions. The training examples are then employed to learn how confident the classifier is for the eHow *Answer* to satisfy the information need of a *Q_new* from Yahoo! Answers.

6.2 Previous Work on Procedural Intent

Only a few studies have investigated procedural intent. Yin et al. [84] presented a two-stage framework for answering *how-to-questions*. The first stage is very similar

to ours, which returns the most similar documents (while we return the most similar questions). However, in the second stage their answers are classified in terms of procedurality (the proportion of procedural text the document contains), while in our framework the answers are classified according to whether they can satisfy the information need of the new question. The work most similar to ours is [72] in which Shtok et al. attempted to resolve unanswered questions in Yahoo! Answers by reusing the repository of past resolved questions. However, due to the inhomogeneous nature of the CQA sites (for example, many questions seek sympathy from other people rather than a question solution), the quality of the answer cannot be guaranteed. In contrast, our approach aims to automatically generate answers from external resources, where all the questions are resolved by the well-formatted procedural instructions. Furthermore, the question context (e.g., the categories where the question was posted) features are completely ignored in [72], which we find plays an important role for the classifier's performance.

6.3 Research Problems Pertaining to Procedural Intent

In this chapter, we define how-to-questions as those whose answer is a set of procedures for achieving a specific goal. A *how-to-question* is typically introduced by the interrogative "how", as presented in Table 6.1, and it can be seen that more than 90% of *how-to-questions* start with "how". However, it is worth noting since "how" has several other usages and many of which are not related to procedural intent. For instance, *How old are you?*, which can usually be satisfied with a simple numeric answer; or *How did John die?*, which is used to know the causes or the circumstances of a certain event and thus is not a procedural use of "how".

This suggests that the presence of "how" cannot be regarded as the sole

Table 6.1: The pattern distribution of how-to-questions over Pets, Health, and Travel categories

Pattern	Pets	Health	Travel
how to	0.553	0.316	0.496
how do/does	0.128	0.506	0.347
how can	0.238	0.113	0.074
is it	0.025	0.039	0.044
what to	0.056	0.026	0.040

indicator of a *how-to-question*. To address this problem, we learned several useful patterns in order to extract *how-to-question* from CQA automatically. Table 6.1 shows the distribution of the top 5 *how-to-questions* patterns that we found in Yahoo! Answers over the *Pets, Health*, and *Travel* categories. It is interesting to notice that the distribution of "how to" pattern is significantly lower in the *Health* category than the other two. We believe that it is probable that people are more prone to have empathy with other people with a "how do you" fashion of enquiry when asking health-based questions .

6.4 Approach to Dealing with Procedural Intent

To begin with, our algorithm retrieves and ranks the similar eHow questions to the new question of Yahoo! Answers. In the second stage the algorithm assesses the effectiveness of the eHow answers (of the most similar questions) for satisfying the information need of the new question. The details regarding this two-stage approach are described below.

6.4.1 Stage One: Top Candidate Selection

An eHow question comprises two parts: a short title and a long body describing the question. However, descriptive texts are not involved in the similarity computation because we find they are usually detrimental rather than beneficial to the search performance.

6.4.1.1 Classic Language Model

Using the classic (query-likelihood) language model [40] for information retrieval , we can measure the relevance of an archive question d with respect to a query question q as:

$$P_{cla}(q|d) = \prod_{w \in q} P_{cla}(w|d) \tag{6.1}$$

$$P_{cla}(w|d) = \frac{Q(w) + m \times P(w|C)}{|Q| + m} \tag{6.2}$$

assuming that each term w in the query q is generated independently by the unigram model of document d. The probabilities $P_{cla}(w|d)$ are estimated from the bag of words in document d with Dirichlet prior smoothing [40], $D(w)$ is the count of word w in question q, C is the whole archive question collection, m is a xed value and is usually determined empirically, $|Q|$ is the total number of word occurrences in Q.

6.4.1.2 Translation-based Language Model

To retrieve and rank the most similar archived eHow questions to the new question from Yahoo! Answers, we adopt the framework similar to [82] which has been demonstrated to be effective for addressing words mismatch problem.

$$P_{tra}(q|d) = \prod_{w \in q} P_{tra}(w|d) \tag{6.3}$$

$$P_{tra}(w|d) = \sum_{t \in d} P(w|t)P(t|d) \tag{6.4}$$

where $P(w|t)$ represents the probability of a document term t being translated into a query term w. As in [82], we estimate such word-to-word translation probabilities $P(w|t)$ on a parallel corpus that consists of 200,000 archived question-answer pairs from Yahoo! Answers.

To exploit evidences from different perspectives for question retrieval, we can mix the above language models via the linear combination [82]:

$$P_{mix}(q|d) = \alpha P_{cla}(q|d) + \beta P_{tra}(q|d) \tag{6.5}$$

where α and β are two non-negative weight parameters satisfying $\alpha + \beta = 1$.

6.4.2 Stage Two: Top Candidate Validation

At this stage, we assess the validity of whether the answer derived from stage-one can satisfy the information need of a new question.

We consider each triplet $< Q\_new, Q\_external, Answer >$ as a new instance of three entities, where entity Q_new denotes a new question from Yahoo! Answers, entity $Q\_external$ is the top candidate question selected from stage-one, and entity $Answer$ is the answer corresponding to the $Q\_external$. Features deriving from the triplets are divided into two types: features which measure the quality of the entity and the features which capture different aspects of similarity between any two entities.

Taken as a whole, we extracted 33 features using a broad range of techniques spanning from query quality assessment to search list validation techniques. We next detail these features.

6.4.2.1 Surface Text Features

Surface Text Statistics: The text features used in the classifier include: text length, maximal IDF within all terms in the text, minimal IDF, average IDF, and

average TF×IDF. These features are capable of identifying and capturing the focus and complexity of the text.

Surface Text Similarity: Features along these lines measure how similar two entities are in terms of lexical overlap, which are computed using the cosine similarity between the TF×IDF weighted word unigram vector space models for any two entities. We measure the the similarity score of $(Q_{new}, Answer)$, of $(Q_{new}, Q_{external})$, and of $(Q_{external}, Answer)$.

6.4.2.2 Question Context Features

Question Asker Statistic: This feature set largely reflects the quality of the asker, such as total number of answers given by the asker and total number of questions posted by the asker.

Question Heuristic Statistic: Features along these lines explore the informativeness of the Q_{new}, including submission time(hour of day), weekdays/weekend, number of answers, and length of best answer.

Topic Similarity: The assumption for the topic categories feature is that if two entities are on the same topic then there is a higher probability that these two entities have the same intent. These features have the power in estimating the similarity of the question topics. For a new question in Yahoo! Answers, we extract the higher-level question category, i.e., *Pets, Health,* and *Travel*, as well as the lower level question category, such as *Birds, Dogs* and *Cats*. One can find that the taxonomy of eHow is in accordance with Yahoo! Answers over these three categories so that we can introduce higher-level and lower-level topic similarities as two boolean features based on the consistency of the categories between the Q_{new} and the $Q_{external}$.

6.4.2.3 Query Feedback Features

The core idea behind Query Feedback is that informational similarity between two questions can be gauged by the similarity between their ranked search result lists. This feature set is the yardsticks measuring both the entity quality and the information need agreement of any two entities.

The following features are considered in our model:

- *Intra-question similarity:* $sim(Q_{title}, Q_{title+body})$, which capture the coherence of a question by identifying when the question title has little in common with its body.
- *Inter-question similarity:* $sim(Q_{new}, Q_{external})$, which addresses the agreement on information need between the two questions.
- *Question-answer similarity:* $sim(Q_{new}, Answer)$, $sim(Q_{external}, Answer)$, which addresses the agreement on information need between question and answer.

As shown in Equation (6.6), the similarity function $sim(q, q')$ is calculated by the *M Measure* [6] which differs from other correlation coefficient methods, such as *Spearman's rank correlation coefficient*[1], in that it gives a higher weight to higher ranking questions, since this measure is based on the intuition that similar ranking of the top questions is more valuable than that of the lower placed questions.

$$\begin{aligned} sim(q, q') = & \sum_{Z} \left| \frac{1}{rank_q(i)} - \frac{1}{rank_{q'}(i)} \right| \\ & + \sum_{S} \frac{1}{rank_q(j)} - \frac{1}{k+1} \\ & + \sum_{T} \frac{1}{rank_{q'}(j)} - \frac{1}{k+1} \end{aligned} \tag{6.6}$$

[1] http://en.wikipedia.org/wiki/Spearman's_rank_correlation_coefficient

Table 6.2: The metadata features with highest information gain.

metadata feature	information gain
Question context: topic category lower-level similarity	0.3421
Surface text similarities: Q_{new} vs. $Answer$	0.3237
Question context: topic category higher-level similarity	0.3159
Query feedback: Q_{new} vs. $Answer$	0.3157
Query feedback: Q_{new} vs. $Q_{external}$	0.2976
Query feedback: $Q_{external}$ vs. $Answer$	0.2903
Answer length	0.117
Query feedback: title of Q_{new} vs. title and body of Q_{new}	0.1105
Question context: asking experience	0.0982
Surface text similarities: Q_{new} vs. $Q_{external}$	0.0937

where Z is the set of the overlapping questions, $\frac{1}{rank_q(i)}$ is the rank of question i in the questions list returned by q, and $\frac{1}{rank_{q'}(i)}$ is its rank in the q' list (both ranks are defined for questions belonging to Z). In addition, S is the set of documents that appear in the q questions list but not in that of the q', while T is the set of questions that appear in the q' list, but not in the q. Lastly, k is the length of the questions list selected from the top candidates (we consider only the top 10 questions).

For the calculation of *Question-answer similarity*, we follow the intuition that similar answers are associated with similar questions. So we retrieve a list of answers from the eHow answer corpus first (by considering *Answer* as a query), from which we then construct the search list with the questions whose corresponding answer is retrieved.

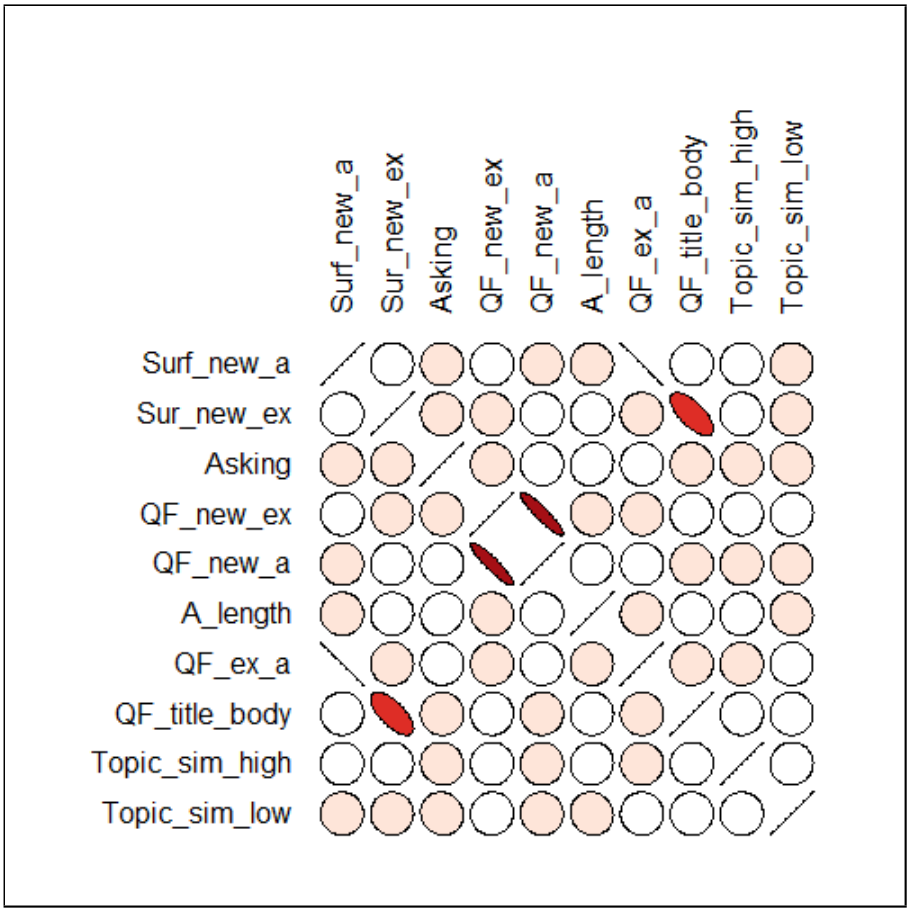

Figure 6.1: Schematic correlation matrix for metadata features reported in Table 6.2

Table 6.2 presents the top 10 features with the highest information gain to the model. It is notable that question context and query feedback features play an important role in identifying the valid answers since they assess the quality of the entities as well as their agreement of information need. Also, topic category features are probably more important than the other ones, the rationale is that they usually represent a distillation of the question subjects and thus are more informative than the other ones for learning the similarity between the question intent. Figure 6.1 displays the correlation matrix regarding the metadata features reported in Table 6.2, which is a visual representation of the relations among those features. In Figure 6.1, each cell is shaded black or blue indicating the polarity of the correlation, and with the intensity of color scaled 0 to 100% in proportion to the magnitude of the correlation. White cells mean the correlation is close to 0, dark red cells mean the correlation is close to -1, and dark blue means the correlation is close to 1. It is clear that most of the features are statistically uncorrelated. Only 4 feature pairs have significant positive correlation (notice the dark red cells). If two features are highly correlated, then one doesn't add any new information to the other, as it is determined by it. The result implies that it is probably better to remove Q_{new} vs. $Q_{external}$ and Q_{new} vs. $Q_{external}$ features, since they have a high correlation with other features.

6.5 Experiments on Procedural Intent

To this end, we implement a QA system based on our two-stage model, which is reported in the following sections.

6.5.1 Experimental Setup

We use two datasets for experiments namely Yahoo! Answers and eHow. The Yahoo! Answers dataset is the *Yahoo! Answers Comprehensive Questions and Answers (v1.0)* corpus, which is kindly provided to the research community by Yahoo! Research through their Webscope[2] programme. The original Yahoo! Answers corpus consists of 4,483,032 questions and their corresponding answers, from which we randomly sampled 1,500 *how-to-questions* from the *Pets, Health,* and *Travel* categories of Yahoo! Answers dataset by using the patterns mentioned in Table 6.1. These questions are submitted to the stage-one system (act as dummy new questions of CQA) to form the triplet (see Section 6.4.2) for feeding the classifier validating the answer. After removing the questions whose probability in the stage-one are smaller to 0.85, finally 1223 questions comprise the triplets dataset for classification: 625 triplets are labelled as positive (relevant) and the other 598 ones are labelled as negative (irrelevant).

The eHow dataset is crawled from the *Pets & Animals, Family Health, Healthcare, Healthy Living, Mental Health, US Travel*, and *Vacations & Travel Planning* categories of the eHow site, dating from 01/09/2012 to 25/04/2013. After removing duplicate questions, there are 253,023, 273,450, and 348,023 examples correspond to the *Pets, Health,* and *Travel* categories, respectively. The surface text features of a question are extracted from the bag-of-words content of the question title after standard pre-processing steps (tokenization, lower-casing, stopword-removal, and stemming) [53].

The performance measurement for classification is the F_1 score, which is the harmonic mean of precision P and recall R. The details regarding the F_1 score has been described in Chapter 3, Section 3.5.2.

[2] http://webscope.sandbox.yahoo.com

Table 6.3: Results of the 10-fold cross validation on the labelled Yahoo! Answers Dataset

Summary of Stratified 10-fold cross-validation	
Correctly Classified Instances	73.3%
Incorrectly Classified Instances	26.7%
Kappa Statistic	0.4545
Mean Absolute Error	0.2799
Root Mean Squared Error	0.3564
Total Number of Instances	1223

6.5.2 Experimental Results

We use the SVM implemented by Platt et al. [63] with a probabilistic output and adopt a linear kernel for this task. The setting of the classifier is the same to that of Chapter 3, Section 3.5.3.

We report the 10-fold cross validation regarding the performance of stage-two in Tables 6.3 and 6.4. Kappa Statistic is a chance-corrected measure of agreement between the classifications and the true classes. It's calculated by taking the agreement expected by chance away from the observed agreement and dividing by the maximum possible agreement. Specifically speaking, it is used in assessing the degree to which two or more raters, examining the same data, agree when it comes to assigning the data to categories. Suppose each object in a group of M objects is assigned to one of n categories. The categories are at nominal scale . For each object, such assignments are done by k raters. The kappa measure of agreement is the ratio:

Table 6.4: The classification accuracy with different feature set (statistical significance using t-test: ** indicates p-value < 0.01 while * indicates p-value < 0.05).

feature set	F_1
surface text	0.616
query feedback	0.436
question context	0.587
surface text + query feedback	**0.653**
surface text + question context	**0.726***
all features	**0.733****

$$K = \frac{P(A) - P(E)}{1 - P(E)} \tag{6.7}$$

where P(A) is the proportion of times the k raters agree, and P(E) is the proportion of times the k raters are expected to agree by chance alone.

The Mean Absolute Error, on the other hand, is a quantity used to measure how close forecasts or predictions are to the eventual outcomes[3].

The mean absolute error is given by:

$$\mathrm{MAE} = \frac{1}{n}\sum_{i=1}^{n}|f_i - y_i| = \frac{1}{n}\sum_{i=1}^{n}|e_i|\,. \tag{6.8}$$

As the name suggests, the mean absolute error is an average of the absolute errors $e_i = |f_i - y_i|$, where f_i is the prediction and y_i the true value. Note that alternative formulations may include relative frequencies as weight factors. Mean Squared Error is the average of the squares of the Mean Absolute Error.

[3] http://en.wikipedia.org/wiki/Mean_absolute_error

The baseline approach used in the experiment is the classifier constructed with both the surface text and query feedback features. While the classification on the basis of topic category or query feedback features alone can only achieve a mediocre performance, it is clear that the combination of the surface text, query feedback, and question context features leads to an approximately 8% performance gain compared to the combination of only the surface text and query feedback features. This suggests that the classifier gains significant insight by incorporating the question context features.

6.6 Summary

The main contribution of this chapter is to show the usefulness of the two-stage model for answering new *how-to-question* in CQA, by leveraging the external resource, i.e., eHow. Our two-stage model supersedes that of the existing one [72], since we employ a more sophisticated retrieval model in stage-one which can address the lexical mismatch problem, since we model the question context (e.g., the categories where the question was posted), in addition to the question text and query feedback. Moreover, the lists similarity is compared by the correlation coefficient, i.e., *M Measure*, which incorporates the ranking of the question lists, instead of the simple counting of the questions overlap that used in [72].

Chapter 7

Understanding User's Causal Intent

In CQA, there are many complex social ecosystems reflecting public opinions, which could allow users to make informed decisions before the final purchase (e.g., they can get a comprehensive review before buying a certain mobile phone). *Why-questions* are particularly important type because their answers often portray the relationship between the product features (the cause) and the user's opinions (the effect), which leads to new challenges raised by sentiment-sensitive applications, compared with those that have proliferated in the traditional fact-based analysis. To answer questions effectively and efficiently, this chapter propose to answer a new *why-question* by making use of the past archived questions. The rest of this chapter is organised as follows. In Section 7.1, we introduce the overview of causal intent. In Section 7.2, we review the related work. In Section 7.3, we define *why-questions* and identified several patterns to extract them from Yahoo! Answers. In Section 7.4 , we introduce the two-stage framework for answering *why-questions*, and we investigate the usefulness of various features. In Section 7.5, we describe the experimental setup and present the experiment results. In Section 7.6, we make

conclusions of this chapter.

7.1 Overview of Causal Intent

Answering *why-questions* is a difficult task since they often encompass insubstantial statement which entails a fairly deep intent analysis of the question context. For example, when one asks the question " Why would anyone buy an iPad?", the asker could either be understood as he/she wants to complain about the product or he/she want to search for some positive reviews about it, the best answer is largely determined by the question-intent orientation (subjective vs. objective). Furthermore, given a good understanding of the question-intent orientation, identifying the best answer for *why-questions* often requires analysis of the user sentiment. For example, there is no standard solution when asking the question "Why is the ipad so expensive?", the answer "apple will launch 3D streams then those people will go nuts" is probably a better choice than "when the other version comes out it will be better and probably cheaper", since the question and the former answer share a similar sentiment.

To study the potential effectiveness of using past questions, to answer a new *why-question*, we carry out our analysis on some active categories of Yahoo! Answers, namely*Consumers Electronics*. More specifically, we extract a subset of *why-questions* asked in these categories in 2006, and validate whether they have a good match from past questions (indicated by a probability above 0.85 produced by the translation-based language model). The $< Q\_new, Q\_past, Answer >$ triplets are then employed to learn how confidence the classifier it is for the *Answer* to satisfying the information need of a Q_new from Yahoo! Answers.

7.2 Previous Work on Causal Intent

Only a few studies have investigated *why-questions*. Girju et al. [24] proposed the first framework for detecting causal relationship from documents. Pechsiri et al. [62], later on, developed a more advanced framework. Unlike the previous frameworks, which only looked at one cause and its corresponding effect,their framework is capable of capturing multiple causes and multiple effects. Oh et al. [60] presented the first framework, which uses sentiment analysis, for improving the why-question classification. However, their work is limited to the NTCIR 6 corpus, which is in Japanese. Verberne et al. [75] experimented with a number of learning models, such as Logistic Regression, Ranking SVM, and SVM map, in differing settings. They reported that their boosting classifier, which blends several classifiers, achieves the best performance.

7.3 Research Problems Pertaining to Causal Intent

In this chapter, we define why-questions as those whose answer is a causative description. Since in this work the focus is on the methods that seek to address sentiment-sensitive applications, we restrict our analysis to the *Consumers Electronics* category where more than 90% of questions involves sentiment orientation.

The distribution of the top 10 question patterns are displayed in Table 7.1. A *why-question* is typically introduced by the interrogative "why", 73% of *opinionated-why-questions* start with the explicit pattern "why". However, 27% of *why-questions* are introduced by implicit patterns, such as $< NP_1$ verb $NP_2 >$, such that the syntactic pattern indicates a causation relationship. Here, we adopt the implicit patterns identified in [24], which revealed 61 "why" patterns. Notice

Table 7.1: The pattern distribution of why-questions over the *Consumer Electronics* category in Yahoo! Answers

Pattern	percentage
why is/are/was	0.273
why do/does	0.228
$< NP_1$ make $NP_2 >$	0.166
why my/your	0.138
why I/you	0.055
$< NP_1$ cause $NP_2 >$	0.039
why don't/doesn't/wouldn't	0.036
$< NP_1$ start $NP_2 >$	0.027
$< NP_1$ related to $NP_2 >$	0.023
$< NP_1$ bring $NP_2 >$	0.014

that we manually sifted out all the implicit patterns, since some of them express a causation relation only in a particular context and only between specific pairs of nouns.

7.4 Approach to Dealing with Causal Intent

To begin with, our algorithm retrieves and ranks the similar past questions to the new question of Yahoo! Answers. Then, in the second stage, the algorithm selects the best answer from the past similar question collection. Lastly, in the third stage, the algorithm assesses the effectiveness of the answer (to the most similar questions)

for satisfying the information need of the new question. The details regarding the three-stage approach are described below.

7.4.1 Stage One: Top Candidate Selection

7.4.1.1 Question Classification

To separate questions that contain opinions from question that enquiry mainly facts, we applied SVM. This approach assumes the availability of a question corpus with pre-assigned opinion and fact which labels at the question level. We randomly sift out 30000 questions from *Consumers Electronics* category, from which 1200 ones are selected as *why-question* using the filter mentioned in Section 7.3. These selected *why-questions* are then manually labelled as either subjective or objective to form the classification training dataset. Eventually, we have 508 subjective and 313 objective ones.

Although SVM can be outperformed in text classification tasks by other methods such as random forests, Li [43] report similar performance for SVM for a similar task, that of distinguishing between subjective and objective content at the question level.

7.4.1.2 Language Model

Using the classic (query-likelihood) language model [86] for information retrieval, we can measure the relevance of an archive question with respect to the given query question. The details for this model can be found in Chapter 6, Section 6.4.1.1. We also adopt the framework similar to [82], which has been demonstrated to be effective for addressing words mismatch problem. The details for this model has been described in Chapter 6, Section 6.4.1.2.

To exploit evidences from different perspectives for question retrieval, we can

mix the above language models via linear combination:

$$P_{mix}(q|d)=\alpha P_{cla}(q|d) + \beta P_{int}(q|d) \tag{7.1}$$

where α and β are two non-negative weight parameters satisfying $\alpha + \beta = 1$.

7.4.2 Stage Two: Top Candidate Validation

Having obtained the top N candidates from stage-one (See Chapter 3, Section 3.5), in stage-two we assess the validity of whether the answer derived from stage-one can satisfy the information need of a new question. We consider each triplet $< Q\_new, Q\_past, Answer >$ as a new instance of three entities, where entity Q_new denotes a new question from Yahoo! Answers, entity Q_past is the top candidate question selected by stage-one, and entity $Answer$ is the instruction text corresponding to the Q_past. Features derived from the triplets are divided into two types: features which measure the quality of the entity and the features which capture different aspects of similarity between any two entities.

Taken as a whole, we extracted 45 features using a broad range of techniques spanning from sentiment analysis and query quality assessment, to search lists validation techniques. We base this decision on the number and strength of sentiment oriented words (either positive or negative), as well as the lexical match of the questions in the sentence. We first discuss how sentiment words are identified by our system, and then we describe the method which aggregates the word sentiment across the question.

7.4.2.1 Sentiment Analysis Features

In product review sites, most questions are opinionated regarding a certain feature of the product. Understanding the question sentiment can help the system to answer the question more effectively and efficiently. In this work, question polar-

ity is identified by using pSenti [56] , which is a concept level sentiment analysis framework. Each sentiment word in pSenti is assigned two numeric scores: Pos(s), and Neg(s), indicating the probabilities of being emotionally Positive and Negative respectively. Since we measure the orientation across an entire sentence or phrase, we used the average per word log-likelihood scores to capture the question polarity. To simplify the task, we presume that there is an overall opinion held by a single asker and is about a single object.

Sentiment Polarity Statistics: The sentiment statistic features used in the classifier include: average per word log-likelihood scores, maximal score within all terms in the text, minimal score, and average score. These features captures the sentiment polarity of the opinion holder.

Sentiment Polarity Similarity: The intuition for this feature set is that if two entities share a similar sentiment polarity then there is a higher probability that these two entities have the same opinion. The features of this line have the power in estimating the similarity between the question sentiment. For example, when asking "If the iPod Mini was so popular, why did Apple stop making it?", reasons with negative sentiment are more desirable than reasons with positive ones to the asker. We introduce sentiment similarities as three boolean features, s_1, s_2, and s_3, based on the sentiment consistency between the Q_{new} and Q_{past}, the Q_{new} and $Answer$, and the Q_{past} and $Answer$. Specifically, $s_1 = 1$ if Q_{new} and Q_{past} have the same sentiment polarities, otherwise $s_1 = 0$. The same rules applied to the other two features as well.

7.4.2.2 Lexico-syntactic Features

One of the concerns is that two questions may share a high syntactical similarity but describe different products. To ensure that the Q_{new} and Q_{past} concern themselves about the same product, we parse each question using the Stanford dependency

parser, from which we extract the main predicate and its arguments, namely the main noun, the main verb and its subject. For example, from "Why shouldn't I buy Iphone?", we extract "buy" as the negated predicate and "Iphone" as the subject. We then test the mismatch features, l_1 and l_2, between the main verb in Q_{new} and Q_{past}, and the mismatch between the subjects of the two. Specifically, $l_1 = 1$ if Q_{new} and Q_{past} have the same verb, otherwise $l_1 = 0$. The same rules applied to l_2 as well. These features help the system to gain insight of semantic inconsistencies between questions. For example, they help in identifying that "Why shouldn't I buy Iphone?" and "Why shouldn't I buy Surface-Pro" have different information needs even though the semantic similarity and text similarity are high.

7.4.2.3 Surface Text Features:

Surface Text Statistics: The text features used in the classifier include: text length, maximal IDF within all terms in the text, minimal IDF, average IDF, and average TF×IDF. These features are capable of revealing the focus and complexity of the text.

Surface Text Similarity: The features of this line measure how similar two entities are in terms of lexical overlap, which are represented by the cosine similarities between the TF×IDF weighted word unigram vector space models for any two entities. We measure the the similarity score of $(Q_{new}, Answer)$, of $(Q_{new}, Q_{external})$, and of $(Q_{external}, Answer)$.

7.4.2.4 Question Context Features

Question Asker Statistic: This feature set largely reflects the quality of the asker, such as total number of answers given by the asker and total number of questions posted by the asker.

Question Heuristic Statistic: The features of this line explore the informative-

Table 7.2: The metadata features with highest information gain.

metadata feature	information gain
Lexico-syntactic: main noun mismatch	0.7375
Surface Text Similarity: Q_{new} vs. *Answer*	0.5822
Query Feedback: Q_{past} vs. *Answer*	0.5603
Question context: topic category higher-level similarity	0.5586
Query Feedback: Q_{new} vs. *Answer*	0.5585
Lexico-syntactic: main verb mismatch	0.3896
Answer Length	0.3704
Sentiment Analysis: sentiment polarity similarity	0.3448
Surface Text Similarity: Q_{new} vs. Q_{past}	0.3020
Question Context: Asking Experience	0.2982

ness behind the Q_{new}, including submission time(hour of day), weekdays/weekend, number of answers, and length of best answer.

7.4.2.5 Query Feedback Features

As shown in Chapter 6, Equation (6.6), the similarity function $sim(q, q')$ is calculated by the *M Measure* [6], which has been described in Section 6.4.2.3, Chapter 6.

Feature Selection: The top 10 features of information gain are reported in Table 7.2. While the most salient features are main noun mismatch, surface text similarity, and query feedback similarity, sentiment analysis similarity and asker's experience are also good indicators for causal intent identification. We can view the

sentiment analysis similarity as the agreement of the semantic and sentiment, which may relate to the nature of user intent. Similarly, it is also notable that question context and query feedback features are important factors in identifying the valid answers, since they assess the quality of the entities as well as their agreement of information need. Another interesting result is the presence of the answer length feature, which confirms our hypothesis that the length of the answer may largely reflect its quality.

7.5 Experiments on Causal Intent

To this end, we implement a QA system based on our two-stage model , which is reported in the following sections.

7.5.1 Experimental Setup

Yahoo! Answers dataset is the *Yahoo! Answers Comprehensive Questions and Answers (v1.0)* corpus, which is kindly provided to research communities by Yahoo! Research through their Webscope[1] programme. The original Yahoo! Answers corpus consists of 4,483,032 questions and their corresponding answers, from which we randomly sampled 1200 *why-questions* from the *Consumer Electronics* category of Yahoo! Answers dataset by using the patterns mentioned in Table 7.1. These questions are submitted to the stage-one system (act as dummy new questions of CQA) to form the triplet (see Section 7.4.2) for feeding the classifier validating the answer. After removing the questions whose probability in the stage-one are smaller than 0.85. Finally, 1000 questions comprise the triplets dataset for classification: 537 triplets are labelled as positive (relevant) and the other 463 ones are labelled as negative (irrelevant). The performance measure for classification is F_1 score, which

[1] http://webscope.sandbox.yahoo.com

Table 7.3: Results of the 10-fold cross validation on the labelled Yahoo! Answers Dataset

Summary of Stratified 10-fold cross-validation	
Correctly Classified Instances	73.9%
Incorrectly Classified Instances	26.1%
Kappa Statistic	0.2239
Mean Absolute Error	0.186
Root Mean Squared Error	0.3647
Total Number of Instances	1223

is the harmonic mean of precision P and recall R. The details regarding the F_1 score has been described in Chapter 3, Section 3.5.2.

7.5.2 Experimental Results

Stage-two Classification: We use the SVM implemented by Platt et al. [64] with a probabilistic output and adopt a linear kernel in this task (See section). The setting of the classifier is the same to that of Chapter 3, Section 3.5.3.

We report the 10-fold cross validation regarding the performance of the stage-two in Table 7.3 and 7.4. The explanation of Kappa Statistic, Mean Absolute Error, Root Mean Squared Error can be found in Chapter 6, Section 6.5.2. The baseline approach used in the experiment is the classifier constructed with both the surface text and query feedback features. While the classification on the basis of topic category or query feedback features alone can only achieve a mediocre performance, it is clear that the combination of the surface text, query feedback, and sentiment similarity features leads to an approximately 10% performance gain compared to the

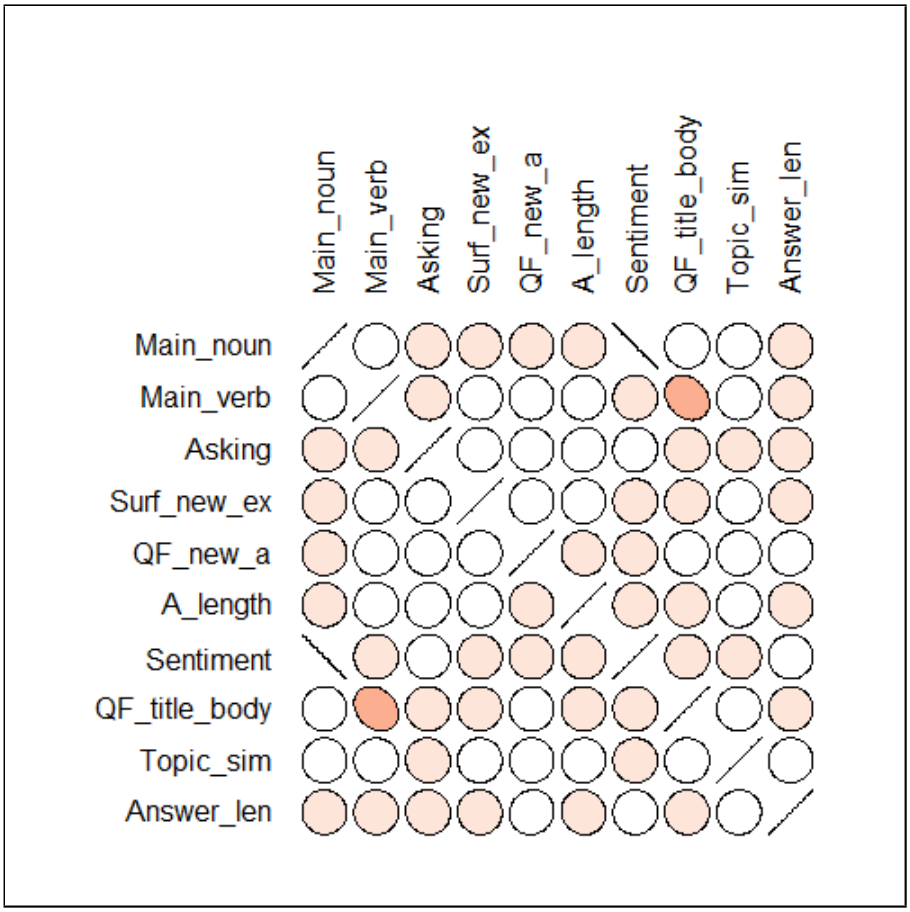

Figure 7.1: Schematic correlation matrix for metadata features reported in Table 7.2

combination of only the surface text and query feedback features. This suggests that the classifier gains significant insight by incorporating the question context features.

The correlation between each features are presented in Figure 7.1. Each cell is shaded red or blue indicating the polarity of the correlation, and with the intensity of color scaled 0 - 100% in proportion to the magnitude of the correlation. White cell means the correlation is close to 0, dark red mean the correlation is close to -1, and dark blue means the correlation is close to 1. It is clear that most of the features are statistically uncorrelated. We did not identify any obvious correlations except for the strong positive correlation between Lexico-syntactic: main noun mismatch and Lexico-syntactic: main verb mismatch.

Feature Ablation: To gain insight of the most important features for this task, we carry out ablation analysis on our feature set. For this, we remove each of the feature categories listed in Section 7.4.2. Table 7.4 presents the F_1 score with each of the feature set removed one by one. When removing lexico-syntactic , query feedback, and sentiment analysis features, the prediction F_1 score drops significantly. On the contrary, question context and surface text features seem to have less effect (they may be redundant provided the presence of the other feature sets). Surprisingly, the asker experience does not seem to be important for predicting user's satisfaction. This may suggest that user's asking experience has little to do with their satisfaction of the question

7.6 Summary

In this chapter, we address the problem of answering *Why-questions* by using the similar framework as Chapter 6. Instead of using external resources to answer a new question, we use the original past questions from the dataset to answer a new question. A series of Natural Language Processing techniques have been

Table 7.4: The SVM classification results of different feature removed (while keeping all the other features intact)

metadata feature	F_1
No Lexico-syntactic: main noun mismatch	0.6375
No Surface Text Similarity: Q_{new} vs. $Answer$	0.6822
No Lexico-syntactic: main verb mismatch	0.6803
No Query Feedback: title of Q_{new} vs. title and body of Q_{new}	0.6886
No Sentiment Analysis: sentiment polarity similarity	0.6985
No Query Feedback: Q_{past} vs. $Answer$	0.6996
No Answer Length	0.7004
No Query Feedback: Q_{new} vs. $Answer$	0.7103
No Surface Text Similarity: Q_{new} vs. Q_{past}	0.7140
No Question Context: Asking Experience	0.7248

employed which includes Stanford dependency parser for computing the lexico-syntactic similarity, and pSenti framework for measuring the sentiment analysis similarity. It was revealed that lexico-syntactic , query feedback, and sentiment analysis features are informative indicators for user's satisfaction of the question.

Chapter 8

Question Retrieval with User Intent

User intent can be exploited for improving many applications in CQA, such as finding similar questions, identifying relevant answers, and recommending potential answerers. This chapter focuses on introducing user intent into question retrieval (i.e., finding similar questions)..

In Section 8.1, we give an overview of question retrieval. In Section 8.2, we review the related work. In Section 8.3, we describe our mixture language modelling approach to question retrieval, and investigate the usefulness of various features. In Section 8.4, we describe the experimental setup and present the experiment results. In Section 8.5, we make our conclusions.

8.1 Overview of Question Retrieval

When a user submits a new question (called a "query") in CQA, the system would usually check whether similar questions have already been asked and answered before, because if so the user's query could be resolved directly.

Finding similar questions in CQA repositories is a difficult task since two questions' user intents may differ significantly even if they bear a close lexical resemblance. For example, at the time of writing this thesis, when submitting the question "Why do people lick their fingers before turning the pages?" to Yahoo! Answers Search, the question "Do you lick your fingers before turning the page?", with a simple best answer "hahaha you been watching lv.. yes i do", is deemed as the best match as these two questions share a significant syntactical similarity. However, the user intents behind these two questions are substantially different: the former one looking for factual knowledge, while the latter one looking for social survey from other people. Hence this chapter will aim to strike a balance between having the question's lexical relevance as high as possible (so that questions with a higher quality and semantic similarity would have a higher rank) and having the question's intent relevance as close as possible (so that questions with a closer intent match would have a higher rank).

8.2 Previous Work on Question Retrieval

The related work of question retrieval can be found in Chapter 2, Section 2.3. The most related to this chapter is the language modelling approach to question retrieval. Jeon et al. [31] designed a retrieval framework based on translational language model to identify similar questions from a large scale archive, but the answer part is ignored in the framework. Liu et al. [82] then proposed a similar approach with question-answer language model, which leverages the relationship within question-answer pairs for additional evidence. Xin et al. [15] examine the usefulness of question-category features for a category-based language model. Our framework is somewhat similar to [15]. However, unlike that previous research which categorise each archive question as either topic relevant or irrelevant, our approach considers each archive question as a mixture of intent with a classifier

output gauging the probability of each category. Moreover, that work only utilises textual features or category features, whereas in our work we also integrate other metadata features.

8.3 Approaches to Question Retrieval

The techniques of language modelling has been previously shown to be effective for question retrieval in CQA.

8.3.1 Classic Language Model

Using the classic (query-likelihood) language model [86] for information retrieval, we can measure the relevance of an archive question with respect to the given query question. The details for this model can be found in Chapter 6, Section 6.4.1.1.

8.3.2 Translation-based Language Model

We also adopt the framework similar to [82], which has been demonstrated to be effective for addressing words mismatch problem. The details for this model has been described in Chapter 6, Section 6.4.1.2.

8.3.3 Intent-based Language Model

There could be different user intents underlying different questions. For example, many questions in CQA are affected by the users' individual interests (empathy, support, and affection, etc.) rather than just informational needs. Here, we propose to take user intent into account for question retrieval in the language modelling

framework:

$$P_{int}(q|d) = \prod_{w \in q} P_{int}(w|d) \tag{8.1}$$

$$P_{int}(w|d) = \sum_{k=1}^{N} P(w|C_k)P(C_k|d) \tag{8.2}$$

where C_k represents a category of user intent, $P(w|C_k)$ is its corresponding unigram language model (See Section 8.3.3.2) and $P(C_k|d)$ is the probability that the document d belongs to that category.

Compared to the category-based language model of Cao et al. [15], the intent-based model above is more general and more robust, because, instead of imposing hard mutually-exclusive classifications, it classifies a question into multiple (user intent) categories with certain probabilities.

8.3.3.1 Probabilistic Classification of User Intent

When computing $P(C_k|d)$ in the above, intent-based language model, we adopt the question taxonomies proposed in Chapter 3, 4, and 5, which classify the user intent of a question as OSS, local/global, and navigational/non-navigational respectively.

In addition to standard textual features (i.e., the bag of words weighted by TF×IDF), a series of metadata features have been identified and exploited for training the probabilistic classifier. We found that question topic, question time, and asker experience are particularly useful for our task of intent-oriented question classification, the details of these features can be found at Chapter 3, Section 3.4, and at Chapter 4, Section 4.4.

8.3.3.2 Estimating Unigram Models for User Intent

Given the probabilistic classification results on all archive questions, we can obtain the unigram language model for each user intent category C_k through maximum-

likelihood estimation:

$$P(w|C_k)=\frac{\sum_{d\in C_k} tf(w,d)P(C_k|d)}{\sum_{w'\in d}\sum_{d\in C_k} tf(w',d)P(C_k|d)} \tag{8.3}$$

where $tf(w,d)$ is the term frequency of word w in document d. It is possible to employ more advanced estimation methods, which is left for future work.

8.3.4 Mixture Model

To exploit evidences from different perspectives for question retrieval, we can mix the above language models via linear combination:

$$P_{mix}(q|d)=\alpha P_{cla}(q|d) + \beta P_{tra}(q|d) + \gamma P_{int}(q|d) \tag{8.4}$$

where α, β, and γ are three non-negative weight parameters satisfying $\alpha+\beta+\gamma = 1$. When $\gamma = 0$, the complete mixture model backs off to the current state-of-the-art approach, i.e., the combination of the classic language model and the translation-based language model only [82].

8.4 Experiments

8.4.1 Experimental Setup

We conducted experiments on two real-world CQA datasets. The first dataset, YA, comes from Yahoo! Answers, which has been explained in Chapter 2, Section 2.7. The second dataset, WA, comes from WikiAnswers. It contains 824,320 questions with their answers collected from WikiAnswers[1] from 2012-01-01 to 2012-05-01.

We first experimented with question classification on 1,539 questions that are randomly selected from the YA dataset and manually labelled according to their user intents. Those questions were split into training and testing sets with a proportion of 2:1.

[1] http://wiki.answers.com/

Table 8.1: The retrieval results using different classifiers. (*indicates 95% confidence level)

	OSS	**Local/Global**	**Navigational/Non-navigational**
MAP(YA)	**0.545**∗	0.512	0.487
P@10(YA)	**0.327**∗	0.269	0.245
MAP(WA)	**0.557**∗	0.544	0.476
P@10(WA)	**0.287**∗	0.265	0.243

8.4.2 Experimental Results

Chapter 3 Section 3.5, Chapter 4 Section 4.5, and Chapter 5 Section 5.4 detailed the performances (miF_1 and maF_1) of question classification via supervised learning and also semi-supervised learning (Co-Training, probability estimation) based on both textual and metadata features. It is clear that semi-supervised learning approaches, which exploit the power of the unlabelled examples, work better than supervised learning approach.

To see which classifier produce the best $P(C_k|d)$ in Equation (8.1) for the performance of intent-based language model, we experiment with different user intent types. The results of different user intents are reported in Table 8.1. It is notable that the retrieval performance on the local/global and navigational/non-navigational are not as good as the OSS one. A possible reason is that question classifications in these two dimensions are more imbalanced than the OSS classification that the power of user intent hasn't been fully exploited by the retrieval yet. Therefore, for the rest of this chapter, we employ OSS as the default user intent taxonomy used in the Equation (8.1). It is possible to combine these three classifiers to achieve even better results, we will explore this in the future work.

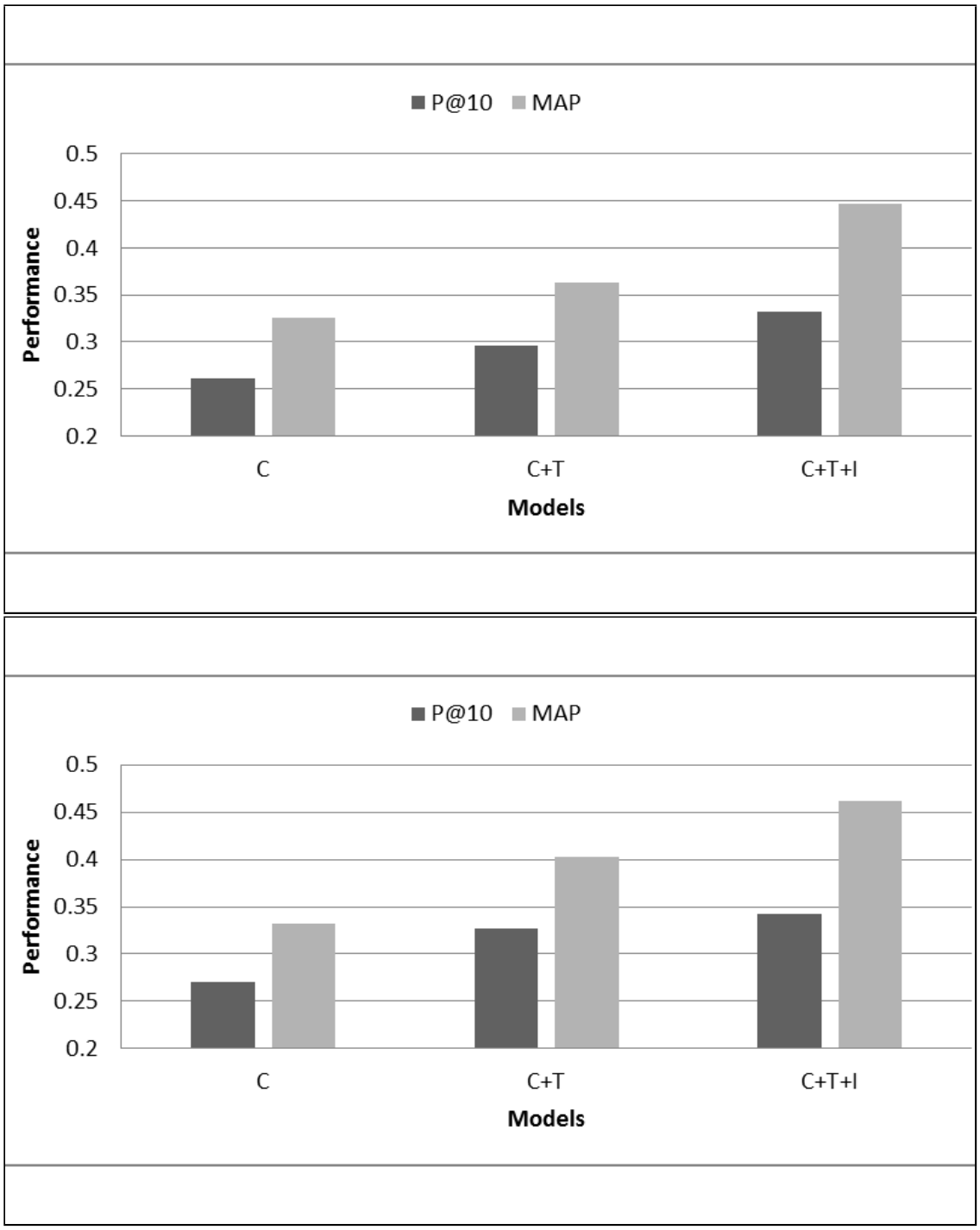

Figure 8.1: The experimental results on Yahoo! Anaswers (up) and WikiAnswers (bottom) respectively

Table 8.2: The model parameters for different question retrieval approaches.

	C	**C+T**	**C+T+I**
α	1	0.3	0.18
β	0	0.7	0.42
γ	0	0.0	0.40

.

We then experimented with question retrieval using a similar set-up as in [82]: 50 questions were randomly sampled from the YA and WA datasets respectively for testing (which were excluded from the CQA retrieval repositories to ensure the evaluation impartiality), and the top archive questions (i.e., search results) returned for each test query question were manually labelled as either relevant or not. In order to see whether user intent relevance can improve question retrieval performance, we compared the following three approaches:

- the baseline approach which only employs the classic language model (C);
- the state-of-the-art approach which combines the classic language model and the translation-based language model (C+T) [82];
- the proposed hybrid approach which blends the classic language model, the translation-based language model, and the intent-based language model (C+T+I).

The model parameters were tuned on the training data to achieve optimal results, as shown in Table 8.2. In the mixture models (C+T) and (C+T+I), the ratio between parameter values α and β was same as that in [82].

The retrieval performances of those approaches, measured by Precision at 10 (P@10) [53] and Mean Average Precision (MAP) [53] , are reported in Figure 8.1. Consistent to the observation in [82], adding the translation-based language model (C+T) brings substantial performance improvement to the classic language

model (C). More importantly, it is clear that our proposed hybrid approach incorporating the intent-based language model (C+T+I) outperforms the state-of-the-art approach (C+T) significantly, according to both P@10 and MAP on YA and WA.

8.5 Summary

The main contribution of this chapter is twofold. First, even though translation-based language model and category-based language model have been investigated independently for CQA just recently, our work is the first attempt to combine these two techniques together in a complementary fashion. Second, unlike the previous retrieval techniques that only look at either textual feature or category feature, we identify and exploit a series of metadata features to move forward the category-based language model performance. We demonstrate that a better result can be achieved by striking a good balance on question's intent relevance and lexical relevance.

Chapter 9

Conclusions and Future Work

While the Web keeps growing, CQA services have developed in complexity with the proliferation of the social media. As a result, questions submitted to a CQA service are often ambiguous or colloquial [70] (at least to some extent). In addition to the enormous efficiency challenges caused by the increasing rate of information production and consumption, CQA services should also endeavor to improve their effectiveness. To this end, understanding the user intent underlying each submitted question becomes a challenging task.

Typical CQA services tend to view the question formulation and the retrieval process as a simple, one-dimensional task. However, the user intent behind questions is usually complex and ambiguous, and CQA systems should be designed to support a variety of characteristics rather than a single textual match. In this thesis we have analysed and characterised five dimensions that can be useful for the detection of users' intent. These dimensions are: subjectivity, locality, navigationality, procedurality, and causality. We introduced a novel intent-based framework, which aims to diversify the potential answers, by fully accounting for the possible user intent underlying the input question. By considering the retrieved answers with these user intents, CQA users will have a better chance of receiving answers which

are both of high quality and thematically relevant. In this scenario, we introduced a two-stage framework which can validate an answer by incorporating the possible user intent underlying its corresponding question.

Throughout this thesis, we analyse and exploit a variety of user intents from different angles. Section 9.1 describes our main contributions and the conclusions drawn from the previous chapters. Section 9.2 summarises the conclusion of each chapter. Section 9.3 discusses several directions for future work, based on the results of each chapter.

9.1 Summary of Thesis Conclusion

There are two main contributions of this thesis:

First, we have proposed how to understand the user intent by classifying the question into five dimensions. We are able to attain consistent and significant classification improvements over the state-of-the-art in this area, by making use of advanced machine learning techniques, such as Co-Training and PU-Learning. In addition to the textual features, a variety of metadata features (such as the category where the question was posted to) are used to model a user's intent, which in turn helps the CQA service to perform better in finding similar questions, identifying relevant answers, and recommending the most relevant answerers.

Second, we have validated the usefulness of user intent in two different CQA tasks. Our first application is question retrieval, where we present a hybrid approach which blends several language modelling techniques, namely, the classic (query-likelihood) language model, the state-of-the-art translation-based language model, and our proposed intent-based language model. Our second application is answer validation, where we present a two-stage model which first ranks similar questions by using our proposed hybrid approach, and then validates whether the answer of the top candidate can be served as an answer to a new question by leveraging

sentiment analysis, query quality assessment, and search lists validation.

9.2 Summary of Conclusion for Each Chapter

In this section, we generalize the main conclusions drawn from the experiments of user intents and their corresponding exploitations. In particular, we introduce the background of CQA services in Chapter 1, from which we then formally define the problems tackled in this thesis.

In Chapter 2, we summarize related work on and Community Question Answering: the basics of a CQA service and question classification (Section 2.2); classical approaches for question retrieval (Section 2.3); answer recommendation (Section 2.5), and answer validation (Section 2.4). The chapter closes with a statistical description of the datasets used (Section 2.7), which form the foundation for several experiments conducted in this thesis.

In Chapter 3, we describe objective, subjective, and social intent from a user-centric perspective, for which we classify questions into three categories according to their underlying user intent, as is described in Section 3.3. We reveal that textual features and metadata features are conditionally independent, and each of them is sufficient for prediction purposed. Therefore they can be exploited as two views in the Co-Training process for enhanced question classification, as described in Section 3.4, in order to make use of a large amount of unlabelled questions, in addition to the small set of manually labelled questions. The user intent (objective/subjective/social) is given by a probabilistic classifier which makes use of both textual and metadata features.

In Chapter 4, we explore users' locality intent. In Section 4.3, questions are classified into two categories according to their intent scope: local or global. In Section 4.4 we describe the challenge for this task: manually labelling questions as local or global for training would be costly. Realising that we could find many

local questions reliably from a few location-related categories (e.g., "Travel"), we propose to build local/global question classifiers in the framework of PU-Learning (i.e., learning from positive and unlabelled examples), and thus remove the need for manually labelling questions. Our experiments on real-world datasets (collected from Yahoo! Answers and WikiAnswers), in Section 4.5, show that for this task the probability estimation approach to PU-learning outperforms S-EM (Spy EM) and Biased-SVM.

Chapter 5 analyses navigational intent, in which questions are classified as navigational and non-navigational. In Section 5.3, we define navigational questions as questions that are resolved (or largely explained) by linked web pages (i.e., in the corresponding answers), which are employed as verbose queries to evaluate the performance of search engines (i.e., by considering the associated linked web pages as relevant documents). In Section 5.4, then, we experiment with the process of identifying new navigational questions from CQA, from which we demonstrate that navigational intent detection can be effectively automated by using textual features and a set of metadata features.

In Chapter 6, we describe procedural intent. In Section 6.3, we define how-to-questions as those whose answer is a set of procedures for achieving a specific goal, form which we then captures a series of empirical patterns to identify how-to-questions. In Section 6.4 we estimate the probability whether a new question in CQA, such as Yahoo! Answers, can be satisfactorily answered by the external resource using a two-stage model similar to factual question answering. A broad range of techniques spanning from query quality assessment to search list validation are leveraged to extract features for our model. In Section 6.5, classifiers with the features modelling the question context (e.g., the categories where the question was posted) are compared to those of the surface text and query feedback of the question.

In Chapter 7, we describe causal intent for the use of product review. In Section 7.3, we define why-questions as those whose answer is a causative description, from which we capture a series of empirical patterns to identify why-questions from Yahoo! Answers. In Section 7.4 we estimate the probability whether a new question in CQA can be used to understand users' opinion towards the product. A broad range of subjectivity computational techniques, such as pSenti and Wordnet, are leveraged to extract features for our model.

In Chapter 8 we present a hybrid approach that blends several language modelling techniques for question retrieval, namely, the classic (query-likelihood) language model, the state-of-the-art translation-based language model, and our proposed intent-based language model. The user intent of each candidate question (objective/subjective/social) is given by a probabilistic classifier, which makes use of both textual features and metadata features.

9.3 Direction of Future Work

In this section, we discuss several directions for future research, which are directly derived from the results of this thesis. These directions are categorized in terms of the broad themes of user intent understanding and user intent exploitation.

9.3.1 User Intent Understanding

Since user intents are often very complex, one way to deepen our understanding on user intent is to explore new taxonomies tailored to those intents. For instance, in Chapter 4, questions are categorized as local and global, with the former extending the administrative place types of Yahoo! Placemaker namely, *Country, State, County, Town, and Local Administrative Area.* However, the unique features attached to a local area may be omitted, such as some landmarks in a city. So an

attractive way for further improving the locality taxonomy is by viewing it in another perspective: building up a unique language model for each county, town, and local administrative area. The language model construction may be a trivial and tedious process, but it should be able to bring about additional performance gain. Lastly, the temporality dimension has not been discussed much in this work. With the unprecedented speed of the question production and consumption, the truly urgent questions may get replaced by other more recently posted questions. An intuitive solution is to simply separate questions which need immediate responses from the other regular questions. Urgent questions may also be further broken down into more detailed intent.

Another way to deepen the understanding of user intent is by improving the performance of the semi-supervised learning models. It is worth noting that the heterogeneous CQA environment presents interesting opportunities for extracting metadata features for guiding question classification. Previous work [3] has revealed that question-answer pairs, answer numbers, user experience, and answer ratings are important features for understanding the information need behind a new question. In addition to answering questions and reputation calculation, some other information may also help the performance of question classification. For example, since CQA sites are communities (no matter how loosely they are organised), the inherent structure and interpersonal dynamics within it can also be utilised as the indicators for intent inference. Co-Training and PU-Learning are shown in Chapter 3 and Chapter 4 respectively, for the identification of user intent. We plan to introduce more sophisticated semi-supervised learning algorithms, such as co-EM Support Vector learning [11], to update the current Co-Training model. Also, expanding the question words using phrase-based features extracted by Latent Dirichlet Allocation (LDA) model would be a promising technique to improve the classification performance.

9.3.2 User Intent Exploitation

User intent can be exploited in many applications, such as question classification, question retrieval, and answer validation, which have been explored throughout this thesis. For future work that builds on the validation of search engines (by making use of navigational questions) in Chapter 5, we will investigate the best approach to query refinement in search engine queries (query expansion or query reduction). The work of Chapter 5 will also be the foundation for future research of utilising phrase/concept detection techniques for query expansion.

For future work that builds on procedural intent in Chapter 6, we will explore more advanced techniques, which are tailored for procedurality extraction to further improve the understanding of procedural text. Since there are many metadata features available for knowledge mining and text features are usually decomposed into a high dimension, it is necessary to incorporate more advanced boosting approach, which combines the power of several learning models such as Random Forest and Gradient Boosting Machines, to allow the classifier to gain insight from different perspectives.

In addition to these mentioned applications, another attractive application of user intent is answer recommendation. The idea is that forwarding an asker's question to someone who has the same or similar intent to the asker can provide good answer recommendation. We plan to employ the translation model (See Chapter 8) and the LDA topic model (See Chapter 2) to predict the user intent based on the textual features. We also plan to introduce the competition-based networks approach [4] to incorporate users' personal and interpersonal features.

9.4 Final Remarks

This thesis contributes in several dimensions regarding the understanding and exploitation of user intent in CQA. As demonstrated throughout the thesis, the principles underlying the framework are not only technically sound, but also practical for real-world applications. From a research perspective, the generality of the framework leads to the investigation of several dimensions of the intent identification problem, including the following questions:

1. Is the user looking for the factual knowledge? For example, the question "In which country in Africa that was colonized by France did assimilation policy succeed?" seeks for details about a specific event. If so then the question has objective intent.

2. Does the user just want to set up a conversation with some other people in the community? For instance, the question "Do you need a friend to work with in London?" If so then the question has social intent.

3. Is there a geographical scope for the question? For example, users querying for a coffee shop are probably looking for one within walking distance. If there is explicit or implicit constraints behind the question scope, then the question has a local intent.

4. What kind of resource is the user seeking for? (e.g., Web links, video streaming, or just a download)? If the user is looking for a web link then the question has a navigational intent. The intent of other resource types may be exploited in the future work.

5. What kind of content is the user seeking? A procedural text telling the user how to do something, or a causative description to explain a phenomenon? We consider the former as procedural intent and latter as causal intent.

6. Has the question been submitted by other users before? If so then how do we find the most similar ones? This leads to the development of our intent-based question retrieval system.

7. Is it important that the answers originate from trusted experts? Is the history of the answerers an important feature? To answer these two questions, we introduce answer validation system to check the answer's credibility.

These investigations led to the publication of five peer-reviewed conference papers directly related to this thesis. Moreover, as discussed in Section 9.3, this thesis opened up directions for other researchers, who may deploy and extend the intent-based framework for different applications.

Appendix A

Extra Experiments on the Two-Stage Model

As mentioned in Chapter 6 Section 6.4 and Chapter 7 Section 7.4, the two-stage model can validate the answer of procedural and causal question. This appendix contains experimental results of applying the same two-stage model for handling local, and navigational question. In Section A.1, we employ the two-stage framework for validating answers of *local* and navigational questions.

A.1 Experiment Set-up

We randomly sampled 1500 local questions from the Dining Out, Travel, and Local Business categories of Yahoo! Answers, these questions are submitted to the two-stage model (act as dummy new questions of CQA) to form the triplet (see Section 6.4.2) for feeding the classifier validating the answer. There are 1178 questions comprise the triplets dataset for classification: 523 triplets are labelled as positive (relevant) and the other 655 ones are labelled as negative (irrelevant).

With a similar manner as local questions, we sampled 1500 navigational

questions from Yahoo! Answers, which have URLs appeared in their answers. There are 1253 questions comprise the triplets dataset for classification: 632 triplets are labelled as positive (relevant) and the other 621 ones are labelled as negative (irrelevant). The details of the two-stage model set-up and the related features are similar to Chapter 6, Section 6.4.

A.2 Experimental Results

The baseline approach used in the experiment is the classifier constructed with both the surface text and query feedback features. While the classification using topic category or query feedback features alone can only achieve a limited performance, it is clear that the combination of the surface text, query feedback, and question context features leads to a significant performance gain for both local and navigational questions. This suggests that the classifier gains significant insight by incorporating the metadata features. More importantly, it is clear that our proposed two-stage model can accurately predict the quality of the answer, regardless of the question types.

Table A.1: The classification results(F_1 value) with different feature set (statistical significance using t-test: ** indicates p-value < 0.01 while * indicates p-value < 0.05).

feature set	local questions	navigational questions
surface text	0.637	0.672
query feedback	0.358	0.539
question context	0.493	0.565
surface text + query feedback	**0.647**	**0.684**
surface text + question context	**0.736***	**0.782***
all features	**0.741****	**0.793****

Bibliography

[1] Lada A. Adamic, Jun Zhang, Eytan Bakshy, and Mark S. Ackerman. Knowledge sharing and yahoo answers: everyone knows something. In *Proceedings of the 17th international conference on World Wide Web*, WWW '08, pages 665–674, New York, NY, USA, 2008. ACM.

[2] Eugene Agichtein, Carlos Castillo, Debora Donato, Aristides Gionis, and Gilad Mishne. Finding high-quality content in social media. In *Proceedings of the International Conference on Web Search and Web Data Mining (WSDM)*, pages 183–194, Palo Alto, CA, USA, 2008.

[3] Ashton Anderson, Daniel Huttenlocher, Jon Kleinberg, and Jure Leskovec. Discovering value from community activity on focused question answering sites: a case study of stack overflow. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '12, pages 850–858, New York, NY, USA, 2012. ACM.

[4] Çiğdem Aslay, Neil O'Hare, Luca Maria Aiello, and Alejandro Jaimes. Competition-based networks for expert finding. In *Proceedings of the 36th international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '13, pages 1033–1036, New York, NY, USA, 2013. ACM.

[5] Ricardo A. Baeza-Yates, Liliana Calderón-Benavides, and Cristina N. González-Caro. The intention behind web queries. In Fabio Crestani, Paolo Ferragina, and Mark Sanderson, editors, *SPIRE*, volume 4209 of *Lecture Notes in Computer Science*, pages 98–109. Springer, 2006.

[6] Judit Bar-Ilan, Mazlita Mat-Hassan, and Mark Levene. Methods for comparing rankings of search engine results. *Comput. Netw.*, 50(10):1448–1463, July 2006.

[7] Jiang Bian, Yandong Liu, Eugene Agichtein, and Hongyuan Zha. Finding the right facts in the crowd: Factoid question answering over social media. In *Proceedings of the 17th International Conference on World Wide Web (WWW)*, pages 467–476, Beijing, China, 2008.

[8] Jiang Bian, Yandong Liu, Ding Zhou, Eugene Agichtein, and Hongyuan Zha. Learning to recognize reliable users and content in social media with coupled mutual reinforcement. In *Proceedings of the 18th international conference on World wide web*, WWW '09, pages 51–60, New York, NY, USA, 2009. ACM.

[9] Avrim Blum and Tom Mitchell. Combining labeled and unlabeled data with co-training. In *Proceedings of the 11th Annual Conference on Computational Learning Theory (COLT)*, pages 92–100, Madison, WI, USA, 1998.

[10] Mohamed Bouguessa, Benoît Dumoulin, and Shengrui Wang. Identifying authoritative actors in question-answering forums: the case of yahoo! answers. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '08, pages 866–874, New York, NY, USA, 2008. ACM.

[11] Ulf Brefeld and Tobias Scheffer. Co-em support vector learning. In *Proceedings of the 21st International Conference on Machine Learning (ICML)*, pages 121–128, Banff, Alberta, Canada, 2004.

[12] Andrei Broder. A taxonomy of web search. *SIGIR Forum*, 36:3–10, September 2002.

[13] Li Cai, Guangyou Zhou, Kang Liu, and Jun Zhao. Large-scale question classification in cqa by leveraging wikipedia semantic knowledge. In *Proceedings of the 20th ACM international conference on Information and knowledge management*, CIKM '11, pages 1321–1330, New York, NY, USA, 2011. ACM.

[14] Xin Cao, Gao Cong, Bin Cui, and Christian S. Jensen. A generalized framework of exploring category information for question retrieval in community question answer archives. In *Proceedings of The 19th International Conference on World Wide Web (WWW)*, pages 201–210, Raleigh, NC, USA, 2010.

[15] Xin Cao, Gao Cong, Bin Cui, Christian Søndergaard Jensen, and Ce Zhang. The use of categorization information in language models for question retrieval. In *Proceedings of the 18th ACM Conference on Information and Knowledge Management (CIKM)*, pages 265–274, Hong Kong, China, 2009.

[16] Ben Carterette, Virgil Pavlu, Evangelos Kanoulas, Javed A. Aslam, and James Allan. Evaluation over thousands of queries. In *Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '08, pages 651–658, New York, NY, USA, 2008. ACM.

[17] Ben Carterette and Mark D. Smucker. Hypothesis testing with incomplete relevance judgments. In *Proceedings of the sixteenth ACM conference on Conference on information and knowledge management*, CIKM '07, pages 643–652, New York, NY, USA, 2007. ACM.

[18] Olivier Chapelle, Bernhard Schölkopf, and Alexander Zien, editors. *Semi-Supervised Learning*. MIT Press, 2006.

[19] Long Chen, Dell Zhang, and Mark Levene. Understanding user intent in community question answering. In *Proceedings of the 21st international conference companion on World Wide Web*, WWW '12 Companion, pages 823–828, New York, NY, USA, 2012. ACM.

[20] Hoa Trang Dang, Diane Kelly, and Jimmy J. Lin. Overview of the TREC 2007 question answering track. In *Proceedings of The Sixteenth Text REtrieval Conference (TREC)*, Gaithersburg, MD, USA, 2007.

[21] Charles Elkan and Keith Noto. Learning classifiers from only positive and unlabeled data. In *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, pages 213–220, Las Vegas, NV, USA, 2008.

[22] Tamer Mohamed Elsayed. *Identity resolution in email collections.* PhD thesis, University of Maryland at College Park, College Park, MD, USA, 2009. AAI3372840.

[23] Oren Etzioni, Michele Banko, Stephen Soderland, and Daniel S. Weld. Open information extraction from the web. *Commun. ACM*, 51(12):68–74, December 2008.

[24] Roxana Girju. Automatic detection of causal relations for question answering. In *Proceedings of the ACL 2003 workshop on Multilingual summarization and question answering - Volume 12*, MultiSumQA '03, pages 76–83, Stroudsburg, PA, USA, 2003. Association for Computational Linguistics.

[25] Luis Gravano, Vasileios Hatzivassiloglou, and Richard Lichtenstein. Categorizing web queries according to geographical locality. In *Proceedings of the 12th International Conference on Information and Knowledge Management (CIKM)*, pages 325–333, New Orleans, LA, USA, 2003.

[26] Jiafeng Guo, Gu Xu, Hang Li, and Xueqi Cheng. A unified and discriminative model for query refinement. In *Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*, pages 379–386, 2008.

[27] F. Maxwell Harper, Daniel Moy, and Joseph A. Konstan. Facts or friends?: Distinguishing informational and conversational questions in social Q&A sites. In *Proceedings of the 27th International Conference on Human Factors in Computing Systems (CHI)*, pages 759–768, Boston, MA, USA, 2009.

[28] F. Maxwell Harper, Daphne Raban, Sheizaf Rafaeli, and Joseph A. Konstan. Predictors of answer quality in online q&a sites. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '08, pages 865–874, New York, NY, USA, 2008. ACM.

[29] Jian Hu, Gang Wang, Fred Lochovsky, Jian-tao Sun, and Zheng Chen. Understanding user's query intent with wikipedia. In *Proceedings of the 18th international conference on World wide web*, WWW '09, pages 471–480, New York, NY, USA, 2009. ACM.

[30] Samuel Huston and W. Bruce Croft. Evaluating verbose query processing techniques. In *Proceedings of the 33rd international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '10, pages 291–298, 2010.

[31] Jiwoon Jeon, W. Bruce Croft, and Joon Ho Lee. Finding similar questions in large question and answer archives. In *Proceedings of the 14th ACM International Conference on Information and Knowledge Management (CIKM)*, pages 84–90, Bremen, Germany, 2005.

[32] Jiwoon Jeon, W. Bruce Croft, Joon Ho Lee, and Soyeon Park. A framework to predict the quality of answers with non-textual features. In *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '06, pages 228–235, New York, NY, USA, 2006. ACM.

[33] Yoonjae Jeong and Sung-Hyon Myaeng. Using wordnet hypernyms and dependency features for phrasal-level event recognition and type classification. In *Proceedings of the 35th European conference on Advances in Information Retrieval*, ECIR'13, pages 267–278, Berlin, Heidelberg, 2013. Springer-Verlag.

[34] Vinay Jethava, Liliana Calderón-Benavides, Ricardo Baeza-Yates, Chiranjib Bhattacharyya, and Devdatt Dubhashi. Scalable multi-dimensional user intent identification using tree structured distributions. In *Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval*, SIGIR '11, pages 395–404, New York, NY, USA, 2011. ACM.

[35] Zongcheng Ji, Fei Xu, Bin Wang, and Ben He. Question-answer topic model for question retrieval in community question answering. In *Proceedings of the 21st ACM international conference on Information and knowledge management*, CIKM '12, pages 2471–2474, New York, NY, USA, 2012. ACM.

[36] Pawel Jurczyk and Eugene Agichtein. Discovering authorities in question answer communities by using link analysis. In *Proceedings of the sixteenth ACM conference on Conference on information and knowledge management*, CIKM '07, pages 919–922, New York, NY, USA, 2007. ACM.

[37] Pawel Jurczyk and Eugene Agichtein. Hits on question answer portals: exploration of link analysis for author ranking. In *Proceedings of the 30th annual*

international ACM SIGIR conference on Research and development in information retrieval, SIGIR '07, pages 845–846, New York, NY, USA, 2007. ACM.

[38] In-Ho Kang and GilChang Kim. Query type classification for web document retrieval. In *Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval*, SIGIR '03, pages 64–71, New York, NY, USA, 2003. ACM.

[39] Dan Klein and Christopher D. Manning. Accurate unlexicalized parsing. In *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics - Volume 1*, ACL '03, pages 423–430, Stroudsburg, PA, USA, 2003. Association for Computational Linguistics.

[40] John Lafferty and Chengxiang Zhai. Document language models, query models, and risk minimization for information retrieval. In *Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '01, pages 111–119, New York, NY, USA, 2001. ACM.

[41] Uichin Lee, Zhenyu Liu, and Junghoo Cho. Automatic identification of user goals in web search. In *Proceedings of the 14th international conference on World Wide Web*, WWW '05, pages 391–400, New York, NY, USA, 2005. ACM.

[42] Baichuan Li, Irwin King, and Michael R. Lyu. Question routing in community question answering: putting category in its place. In *Proceedings of the 20th ACM International Conference on Information and Knowledge Management (CIKM)*, pages 2041–2044, Glasgow, Scotland, UK, 2011.

[43] Baoli Li, Yandong Liu, Ashwin Ram, Ernest V. Garcia, and Eugene Agichtein. Exploring question subjectivity prediction in community QA. In *Proceedings*

of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR), pages 735–736, Singapore, 2008.

[44] Xin Li and Dan Roth. Learning question classifiers: the role of semantic information. *Nat. Lang. Eng.*, 12(3):229–249, September 2006.

[45] Shu-Jung Lin and Wen-Hsiang Lu. Learning question focus and semantically related features from web search results for chinese question classification. In *Proceedings of the Third Asia conference on Information Retrieval Technology*, AIRS'06, pages 284–296, Berlin, Heidelberg, 2006. Springer-Verlag.

[46] Bing Liu, Yang Dai, Xiaoli Li, Wee Sun Lee, and Philip S. Yu. Building text classifiers using positive and unlabeled examples. In *Proceedings of the Third IEEE International Conference on Data Mining*, ICDM '03, pages 179–, Washington, DC, USA, 2003. IEEE Computer Society.

[47] Bing Liu, Wee Sun Lee, Philip S. Yu, and Xiaoli Li. Partially supervised classification of text documents. In *Proceedings of the 19th International Conference on Machine Learning (ICML)*, pages 387–394, San Francisco, CA, USA, 2002.

[48] Mingrong Liu, Yicen Liu, and Qing Yang. Predicting best answerers for new questions in community question answering. In *Proceedings of the 11th international conference on Web-age information management*, WAIM'10, pages 127–138, Berlin, Heidelberg, 2010. Springer-Verlag.

[49] Qiaoling Liu, Yandong Liu, and Eugene Agichtein. Exploring web browsing context for collaborative question answering. In *Proceedings of the third symposium on Information interaction in context*, IIiX '10, pages 305–310, New York, NY, USA, 2010. ACM.

[50] Yandong Liu and Eugene Agichtein. On the evolution of the Yahoo! answers QA community. In *Proceedings of the 31st Annual International ACM SIGIR*

Conference on Research and Development in Information Retrieval (SIGIR), pages 737–738, Singapore, 2008.

[51] Yandong Liu, Jiang Bian, and Eugene Agichtein. Predicting information seeker satisfaction in community question answering. In *Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '08, pages 483–490, New York, NY, USA, 2008. ACM.

[52] Yuanjie Liu, Shasha Li, Yunbo Cao, Chin-Yew Lin, Dingyi Han, and Yong Yu. Understanding and summarizing answers in community-based question answering services. In *Proceedings of the 22nd International Conference on Computational Linguistics (COLING)*, pages 497–504, Manchester, UK, 2008.

[53] Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schtze. *Introduction to Information Retrieval*. Cambridge University Press, 2008.

[54] Andrew McCallum and Kamal Nigam. A comparison of event models for naive bayes text classification. In *IN AAAI-98 WORKSHOP ON LEARNING FOR TEXT CATEGORIZATION*, pages 41–48. AAAI Press, 1998.

[55] Eduarda Mendes Rodrigues and Natasa Milic-Frayling. Socializing or knowledge sharing?: Characterizing social intent in community question answering. In *Proceeding of the 18th ACM Conference on Information and Knowledge Management (CIKM)*, pages 1127–1136, Hong Kong, China, 2009.

[56] Andrius Mudinas, Dell Zhang, and Mark Levene. Combining lexicon and learning based approaches for concept-level sentiment analysis. In *Proceedings of the First International Workshop on Issues of Sentiment Discovery and Opinion Mining*, WISDOM '12, pages 5:1–5:8, New York, NY, USA, 2012. ACM.

[57] Kevin Kyung Nam, Mark S. Ackerman, and Lada A. Adamic. Questions in, knowledge in?: a study of naver's question answering community. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '09, pages 779–788, New York, NY, USA, 2009. ACM.

[58] Xingliang Ni, Yao Lu, Xiaojun Quan, Liu Wenyin, and Bei Hua. User interest modeling and its application for question recommendation in user-interactive question answering systems. *Inf. Process. Manage.*, 48(2):218–233, March 2012.

[59] Mark O'Connor, Dan Cosley, Joseph A. Konstan, and John Riedl. Polylens: a recommender system for groups of users. In *Proceedings of the seventh conference on European Conference on Computer Supported Cooperative Work*, ECSCW'01, pages 199–218, Norwell, MA, USA, 2001. Kluwer Academic Publishers.

[60] Jong-Hoon Oh, Kentaro Torisawa, Chikara Hashimoto, Takuya Kawada, Stijn De Saeger, Jun'ichi Kazama, and Yiou Wang. Why question answering using sentiment analysis and word classes. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, EMNLP-CoNLL '12, pages 368–378, Stroudsburg, PA, USA, 2012. Association for Computational Linguistics.

[61] Aditya Pal and Scott Counts. Identifying topical authorities in microblogs. In *Proceedings of the fourth ACM international conference on Web search and data mining*, WSDM '11, pages 45–54, New York, NY, USA, 2011. ACM.

[62] Chaveevan Pechsiri and Asanee Kawtrakul. Mining causality from texts for question answering system. *IEICE - Trans. Inf. Syst.*, E90-D(10):1523–1533, October 2007.

[63] John C. Platt. *Fast training of support vector machines using sequential minimal optimization*, pages 185–208. MIT Press, Cambridge, MA, USA, 1999.

[64] John C. Platt. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. In *Advances in Large Margin Classifiers*, pages 61–74. MIT Press, 1999.

[65] Bo Qu, Gao Cong, Cuiping Li, Aixin Sun, and Hong Chen. An evaluation of classification models for question topic categorization. *J. Am. Soc. Inf. Sci. Technol.*, 63(5):889–903, May 2012.

[66] Mingcheng Qu, Guang Qiu, Xiaofei He, Cheng Zhang, Hao Wu, Jiajun Bu, and Chun Chen. Probabilistic question recommendation for question answering communities. In *Proceedings of the 18th international conference on World wide web*, WWW '09, pages 1229–1230, New York, NY, USA, 2009. ACM.

[67] Anna N. Rafferty and Christopher D. Manning. Parsing three german treebanks: lexicalized and unlexicalized baselines. In *Proceedings of the Workshop on Parsing German*, PaGe '08, pages 40–46, Stroudsburg, PA, USA, 2008. Association for Computational Linguistics.

[68] Fatemeh Riahi, Zainab Zolaktaf, Mahdi Shafiei, and Evangelos Milios. Finding expert users in community question answering. In *Proceedings of the 21st international conference companion on World Wide Web*, WWW '12 Companion, pages 791–798, New York, NY, USA, 2012. ACM.

[69] Eldar Sadikov, Jayant Madhavan, Lu Wang, and Alon Halevy. Clustering query refinements by user intent. In *Proceedings of the 19th international conference on World wide web*, WWW '10, pages 841–850, New York, NY, USA, 2010. ACM.

[70] Chirag Shah and Jefferey Pomerantz. Evaluating and predicting answer quality in community qa. In *Proceedings of the 33rd international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '10, pages 411–418, New York, NY, USA, 2010. ACM.

[71] Yelong Shen, Jun Yan, Shuicheng Yan, Lei Ji, Ning Liu, and Zheng Chen. Sparse hidden-dynamics conditional random fields for user intent understanding. In *Proceedings of the 20th international conference on World wide web*, WWW '11, pages 7–16, New York, NY, USA, 2011. ACM.

[72] Anna Shtok, Gideon Dror, Yoelle Maarek, and Idan Szpektor. Learning from the past: Answering new questions with past answers. In *Proceedings of the 21st International Conference on World Wide Web (WWW)*, pages 759–768, Lyon, France, 2012.

[73] Maggy Anastasia Suryanto, Ee Peng Lim, Aixin Sun, and Roger H. L. Chiang. Quality-aware collaborative question answering: methods and evaluation. In *Proceedings of the Second ACM International Conference on Web Search and Data Mining*, WSDM '09, pages 142–151, New York, NY, USA, 2009. ACM.

[74] Xinhui Tu, Tingting He, Long Chen, Jing Luo, and Maoyuan Zhang. Wikipedia-based semantic smoothing for the language modeling approach to information retrieval. In *Proceedings of the 32nd European conference on Advances in Information Retrieval*, ECIR'2010, pages 370–381, Berlin, Heidelberg, 2010. Springer-Verlag.

[75] Suzan Verberne, Hans Halteren, Daphne Theijssen, Stephan Raaijmakers, and Lou Boves. Learning to rank for why-question answering. *Inf. Retr.*, 14(2):107–132, April 2011.

[76] Ellen M. Voorhees. The trec question answering track. *Nat. Lang. Eng.*, 7(4):361–378, December 2001.

[77] Ellen M. Voorhees. Evaluating the evaluation: a case study using the trec 2002 question answering track. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology - Volume 1*, NAACL '03, pages 181–188, Stroudsburg, PA, USA, 2003. Association for Computational Linguistics.

[78] Kai Wang and Tat-Seng Chua. Exploiting salient patterns for question detection and question retrieval in community-based question answering. In *Proceedings of the 23rd International Conference on Computational Linguistics*, COLING '10, pages 1155–1163, Stroudsburg, PA, USA, 2010. Association for Computational Linguistics.

[79] Kai Wang, Zhao-Yan Ming, Xia Hu, and Tat-Seng Chua. Segmentation of multi-sentence questions: towards effective question retrieval in cqa services. In *Proceedings of the 33rd international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '10, pages 387–394, New York, NY, USA, 2010. ACM.

[80] Pu Wang, Jian Hu, Hua-Jun Zeng, Lijun Chen, and Zheng Chen. Improving text classification by using encyclopedia knowledge. In *Proceedings of the 2007 Seventh IEEE International Conference on Data Mining*, ICDM '07, pages 332–341, Washington, DC, USA, 2007. IEEE Computer Society.

[81] Yunjie Xu, Hee-Woong Kim, and Atreyi Kankanhalli. Task and social information seeking: Whom do we prefer and whom do we approach? *J. Manage. Inf. Syst.*, 27(3):211–240, January 2010.

[82] Xiaobing Xue, Jiwoon Jeon, and W. Bruce Croft. Retrieval models for question and answer archives. In *Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*, pages 475–482, Singapore, 2008.

[83] Yiming Yang and Xin Liu. A re-examination of text categorization methods. In *Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*, pages 42–49, Berkeley, CA, USA, 1999.

[84] Ling Yin. A two-stage approach to retrieving answers for how-to questions. In *Proceedings of the Eleventh Conference of the European Chapter of the Association for Computational Linguistics: Student Research Workshop*, EACL '06, pages 63–70, Stroudsburg, PA, USA, 2006. Association for Computational Linguistics.

[85] Quan Yuan, Gao Cong, Aixin Sun, Chin-Yew Lin, and Nadia Magnenat Thalmann. Category hierarchy maintenance: a data-driven approach. In *Proceedings of the 35th international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '12, pages 791–800, New York, NY, USA, 2012. ACM.

[86] ChengXiang Zhai. *Statistical Language Models for Information Retrieval*. Morgan & Claypool Publishers, 2008.

[87] Dell Zhang and Wee Sun Lee. Question classification using support vector machines. In *Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Informaion Retrieval (SIGIR)*, pages 26–32, Toronto, Canada, 2003.

[88] Guangyou Zhou, Li Cai, Jun Zhao, and Kang Liu. Phrase-based translation model for question retrieval in community question answer archives. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies - Volume 1*, HLT '11, pages 653–662, Stroudsburg, PA, USA, 2011. Association for Computational Linguistics.

[89] Tom Chao Zhou, Michael R. Lyu, and Irwin King. A classification-based approach to question routing in community question answering. In *Proceedings of the 21st World Wide Web Conference (WWW), Companion Volume*, pages 783–790, Lyon, France, 2012.

[90] Zhemin Zhu, Delphine Bernhard, and Iryna Gurevych. A multi-dimensional model for assessing the quality of answers in social q&a sites. In *ICIQ*, pages 264–265, 2009.

[91] Justin Zobel. How reliable are the results of large-scale information retrieval experiments? In *Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '98, pages 307–314, New York, NY, USA, 1998. ACM.

Publications

[1] Long Chen, Dell Zhang, and Mark Levene. Understanding and exploiting user's navigational intent in community question answering. In *Proceedings of the 9th Asia Information Retrieval Societies Conference (AIRS)*, pages 392–403, Singapore, December 2013.

[2] Long Chen, Dell Zhang, and Mark Levene. Question retrieval with user intent. In *Proceedings of the 36th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*, pages 973–976, Dublin, Ireland, July 2013.

[3] Long Chen, Dell Zhang, and Mark Levene. Understanding user intent in community question answering. In *Proceedings of the 21st World Wide Web Conference (WWW), Companion Volume*, pages 823–828, Lyon, France, April 2012.

[4] Long Chen, Dell Zhang, and Mark Levene. Identifying local questions in community question answering. In *Proceeding of the 8th Asia Information Retrieval Societies Conference (AIRS)*, pages 284–290, Tianjin, China, December 2012.

[5] Xinhui Tu, Tingting He, Long Chen, Jing Luo, and Maoyuan Zhang. Wikipedia-based semantic smoothing for the language modeling approach to information retrieval. In *Proceeding of the 32nd European Conference on Advances in Information Retrieval (ECIR)*, pages 370–381, Milton Keynes, UK, March 2010.

Index